Architecture SSM sécurisée utilisant un proxy IGMP

Anis Ben Hellel

Rapport de projet de fin d'études : Architecture SSM sécurisée utlisant un Proxy IGMP

Ben Hellel Anis

15 mai 2002

Table des matières

Table des figures

Chapitre 1

Introduction

1.1 Contexte général

L'évolution rapide des technologies vers les réseaux haut-débit et la vitesse de plus en plus rapide des processeurs ont favorisé le développement de nouvelles classes d'applications. Il s'agit des applications multimédias comme l'audio et la vidéo conférence. Pour ce type d'applications, le trafic est généralement du type multidestinataire. De ce fait, pour ces applications coopératives, la communication de groupe est devenue un concept non seulement important mais nécessaire. La transmission multipoint apparaît comme le moyen le plus efficace pour envoyer des données à un groupe de destinataires au lieu de les envoyer à chaque destinataire séparément, ce qui réduit la bande passante utilisée.

C'est en 1988 que Deering a proposé le premier modèle de communication de groupes, et a défini les premiers pas vers l'exploitation efficace du multicast sur l'échelle des réseaux étendus comme Internet. Les systèmes utilisent un protocole d'adhésion aux groupes, IGMP, pour exprimer leur intérêt pour les groupes. Des protocoles de routage intra-domaine et inter-domaine prennent ensuite en charge l'acheminement du trafic multicast aux membres des groupes. En fait, ce modèle représente le modèle de communications de groupes dynamiques à large échelle.

Cependant, les besoins au controle d'accès et à une manipulation efficace des sources bien connues ont abouti à la définition d'un nouveau modèle de communications muti-

cast. C'est le modèle SSM (Source Specefic multicast). Ce modèle offre la possibilité à un récepteur de choisir en plus des groupes, les sources desquelles il veut recevoir le trafic multicast. L'architecture SSM met en jeu des routeurs multicast classiques implémentant la version 3 du protocole IGMP avec un protocole de routage multicast comme PIM-SM. Cependant, il n'est pas toujours nécessaire d'implanter un protocole de routage multicast dans tous les routeurs. Il suffit de consulter l'information fournie par le protocole IGMPv3 pour décider d'envoyer ou non un paquet sur les différentes interfaces du routeur. C'est le principe de fonctionnement du Proxy IGMP.

Le but de mon stage était d'implémenter un proxy IGMP sous FreeBSD et puis de l'intégrer à une archirecture SSM sécurisée. L'implémentation du proxy IGMP nécessite une implémentation du protocole IGMPv3 coté hote et routeur. Ces implémentations sont disponibles. Celle de IGMPv3 cote routeur a été la squelette sur laquelle a eu lieu mes implémentations.

1.2 Organisation du rapport

Dans le présent rapport, nous présentons en premier lieu l'architecture multicast courante ASM (Any Source Multicast) en révélant ses principeaux problemes. Nous présentons par la suite l'architecture SSM en détaillant ses composants et ses principeaux apports. Dans le troisieme chapitre, nous décrivons l'implementation du protocole IGMPv3-routeur qui représente la squelette sur la quelle a eu lieu l'implementation du proxy IGMP. Dans le quatrieme chapitre, nous spécifions les principales fonctionnalités que doit assurer le proxy IGMP. Le chapitre d'apres propose des détails sur son implémentation. Enfin, le dernier chapitre est une évaluation des performances du Proxy IGMP. Cette évaluation consiste en une comparaison de celui-ci avec un routeur classique intégrant PIM-SM.

1.3 Equipe d'accueil

Ce stage a été effectué au sein du projet RESEDAS commun au LORIA (Laboratoire LOrrain de Recherche en Informatique et ses Applications) et à l'INRIA (l'Institut Natio-

nal de Recherche en Informatique et en Automatique) Lorraine. Les principaux axes de recherche de cette équipe sont l'administration des réseaux, les calculs distribués sur des réseaux de stations hétérogènes, et des travaux autour du protocole Internet de nouvelle génération (IPv6).

Chapitre 2

Le modèle multicast ASM

2.1 Introduction

Le réseau Internet offre trois manières de communication différentes : la communication unicast, broadcast et multicast. Le principe de l'unicast est d'envoyer des paquets IP à un récepteur unique. Cependant le broadcast consiste à diffuser une information à tous les ordinateurs du réseau. Quant au multicast, il permet d'envoyer des données à un groupe de recepteurs. Plutôt que d'envoyer les données vers chacune des machines clientes (unicast), la communication multicast permet de n'envoyer l'information qu'une seule fois et chaque ordinateur client la récupère.

Ainsi, le protocole Internet IPv4 définie trois types différents d'adresses qui sont les adresses unicast, les adresses multicast et les adresses broadcast. Les adresse unicast permettent les communications point à point. L'adresse IP de l'ordinateur destinataire est spécifiée dans les paquets émis. Il existe 3 classes d'adresses unicast : La classe A, B ou C. Cependant, les adresses broadcast sont connus par toutes les machines du même sous-réseau. Chaque paquet envoyé avec une adresse broadcast est reçu et traité par chacun des ordinateurs du sous-réseau comme s'il leur était personnellement adressé. Quant aux communications multicast, elles introduisent la notion d'adresses de groupe. Chaque client multicast s'enregistre avec une adresse de groupe à laquelle les données seront envoyées. Les adresses de groupe multicast sont de classe D. En notation décimale, elles sont comprises

entre 224.0.0.0 et 239.255.255.255.

Dans ce chapitre, nous présentons les éléments principaux de l'architecture de routage multicast ASM (Any Source Multicast) déployée actuellement dans Internet. Nous détaillons le modèle de référence IP multicast, avant de présenter le mécanisme d'adhésion aux groupes et le protocole de routage multicast intra-domaine PIM-SM

2.2 Apports du multicast

La communicatsion unicast permet à un expéditeur d' envoyer des paquets à un récepteur unique. Pendant des années, ce mode de communication a été suffisant dans Internet. Cependant, l'augmentation du trafic multimédia (audio/vidéo) multidestinataire au cours de la dernière décenie aunsi que la grande consommation des flux audio et vidéo en terme de bande passante fait que l'utilisation de l'unicast pour ce genre d'applications induit une surcharge du réseau. Le modèle multicast a ainsi été proposé par Deering afin de résoudre ce problème. La communication muilticast effectue la transmission d'un datagramme IP à un groupe de façon performante. En effet, le principe du multicast est de repousser au plus tard l'inévitable duplication des datagrammes IP. Pour ce faire, une seule copie de chaque datagramme transite entre les différents routeurs et la duplication des datagrammes est prise en charge par les routeurs eux-mêmes (figure 2.1). Le multicast permet donc un routage efficace dans le cas d'une communication de groupe en réduisant énormément la charge du réseau et les délais de transmission.

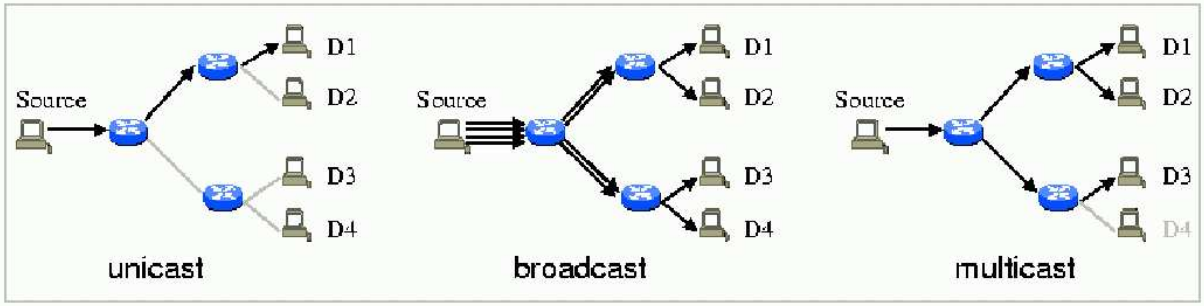

FIG. 2.1 – *Modes de transmission*

D'autre part, le multicast permet d'offrir une transmission efficace quand les adresses des destinataires sont inconnues ou changent régulièrement. l'utilisation des adresses de groupes au lieu des adresses individuelles permet d'atteindre des destinations sans être obligés de connaitre leur adresse exacte. Ceci est très important dans le cas, par exemple, des applications de recherche dans des bases de données réparties ou pour les serveurs web. La communication multicast se base sur le modèle de référence IP Multicast qui sera présenté dans la section suivante.

2.3 Le modèle IP Multicast

Le modèle IP multicast est une extension du protocole IP faisant partie de la suite de protocoles du modèle TCP/IP. Le multicast est basé sur le concept de groupes. Un groupe est un ensemble de stations dont le nombre varie de zéro à l'infini. Il n'a aucune borne physique ou géographique. La composition d'un groupe est dynamique, une station peut joindre un groupe ou quitter un groupe à tout moment. Une station peut envoyer des données à un groupe sans en faire partie. Il exsite des groupes permanents, d'adresses bien connues, qui peuvent avoir un nombre de membres occasionnellement nul. Par contre, d'autres groupes sont temporaires et possèdent des adresses allouées dynamiquement qui n'existent que tant qu'il y a des membres.

Un groupe est identifié pas une adresse de groupe. Les adresses de groupe forment un sous ensemble des adresses IP de clase D dans IPv4 et des adresses avec le préfixe FF00/8 dans IPv6.

Les communications multicast sont gérées par différents protocoles qui mettent en oeuvre le modèle IP Multicast. Ces protocoles sont organisés en couches et sont décrits dans la figure 2.2.

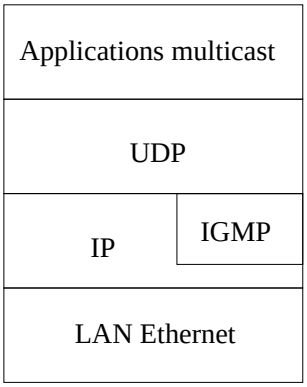

FIG. 2.2 – *Pile protoclolaire de IP Multicast*

2.4 L'architecture multicast ASM (Any Source Multicat)

Tous les réseaux multicast d'aujourd'hui supportent le modèle de service ASM (Any Source Multicast). L'architecture de communication ASM se compose de trois niveaux. Le premier niveau est le modèle de service multicast pour les hôtes. Ce modèle définit les mécanismes permettant à un hôte dee recevoir et transmettre des paquets multicast. Le deuxième niveau contient un protocole de liaison entre les hôtes et les routeurs qui les gèrent, c'est IGMP (Internet Group Management Protocol). Le troisième niveau contient les protocoles de routage qui définissent les mécanismes et les structures suivant lesquels on propage les paquets multicast à travers les réseaux. Les routeurs s'echangent des messages l'un avec l'autre selon un protocole de routage pour construire un arbre de diffusion reliant tous les hôtes. Pour construire les arbres de diffusion, ces protocoles utilisent plusieurs techniques qu'on peut classer en deux groupes: les techniques basées sur les arbres partagés et celles basées sur la source. De nombreux protocoles de routage existent et diffèrent principalement par le type d'arbre de diffusion construit [IPMULTICAST, PIM-ARCH, PIM-SM, PIM-SM-NEW, PIM-DM].

2.4.1 Service multicast

L'envoi et la réception des paquets multicast ne sont pas fondamentalement différents du service IP unicast, à part l'adressage. Toutefois plusieurs modifications sont nécessaires pour qu'un hôte soit capable de transmettre et de recevoir le trafic multicast. Dans le cas où les sources et les destinations se trouvent sur le même LAN, la transmission et la réception multicast restent relativement simples. La source envoie le paquet multicast avec l'adresse du groupe, la carte d'interface réseau fait la correspondance (mapping) entre l'adresse de la classe D et l'adresse multicast dans le LAN (par exemple l'adresse Ethernet). Le système d'exploitation diffuse les paquets de données. Un hôte qui veut recevoir ce trafic, indique à son interface l'adresse correspondante. Il notifit sa couche réseau (en l'occurence IP) qu'il émis sur le lien local, l'ensemble des machines ayant adhéré à ce groupe réceptionne les dits paquets. Les choses se compliquent quand les sources et les récepteurs se trouvent sur des LAN différents, les routeurs sont obligés, dans ce cas, d'implémenter un protocole de routage pour acheminer les données multicast, et d'utiliser le protocole IGMP pour gérer l'adhésion aux groupes d'hôtes sous leurs responsabilité .

Pour envoyer un paquet multicast, trois ajouts sont nécessaires par rapport à l'opération normale IP-Send . Le premier est la définition d'une durée de vie TTL (Time To Live) pour limiter la portée de l'émission du paquet. Le deuxième ajout est la spécification de l'interface réseau sortant en cas d'appartenance à plusieurs LANs. Le troisième concerne les sources qui sont en même temps membres de groupe, elles ont la possibilité d'activer ou désactiver la réception de leurs propres paquets.

2.4.2 IGMP pour ASM

L'Internet Group Management Protocol IGMP est le protocole de communication entre les routeurs multicast et les hôtes multicast dans un même LAN. Il permet à un hôte de s'abonner à un groupe. Ainsi les routeurs posséderont des informations sur les groupes existants dans leur LAN; ces informations seront ultérieurement utilisées par les protocoles de routage multicast pour acheminer les paquets. En cas de présence de plusieurs routeurs multicast pour un même LAN, un seul est élu pour envoyer les messages IGMP de type "query" ou sollicitations: le Routeur Désigné ou Designated Router (DR).

Le Routeur Désigné du LAN interroge périodiquement tous les hôtes en utilisant le message "Query". Le routeur demande ainsi à chaque hôte à quel groupe il veut s'abonner (figure 2.3). Quand un hôte reçoit la sollicitation (IGMP-query), il envoie une réponse (IGMP-report) indiquant les adresses des groupes qui l'intéressent. Si le routeur ne reçoit aucune réponse pour un groupe donné, il arrête l'émission des paquets multicast de ce groupe. Notons que les réponses des hôtes sont entendues par tous les routeurs (DR ou non) et ainsi chaque routeur peut décider d'ajouter ou d'éliminer un groupe.

La figure 2.3 représente le mécanisme d'abonnement dans le protocole IGMP. Trois versions du protocole IGMP se sont succédées. Seules les deux premières versions font partie du modèle ASM.

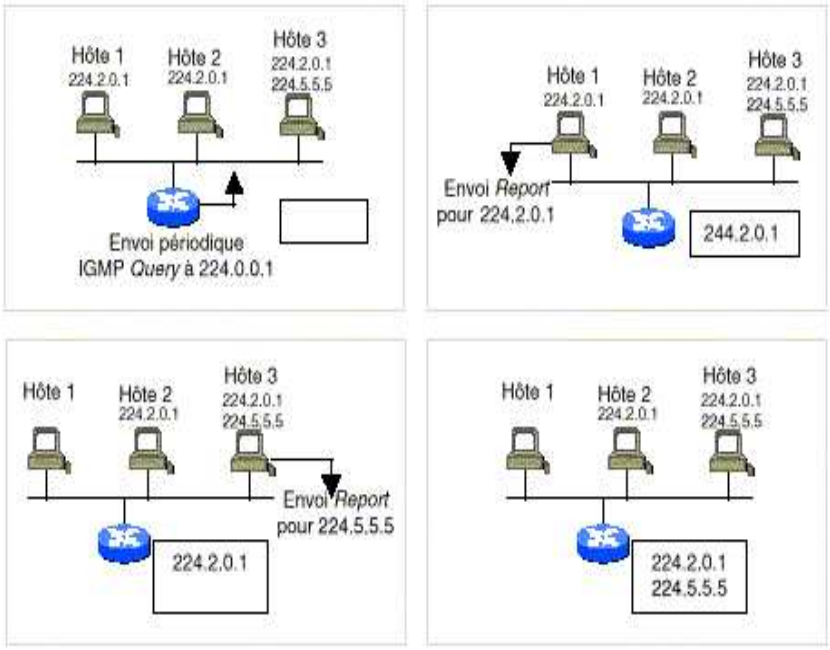

FIG. 2.3 – *Mecanisme d'abonnement à un groupe dans IGMP*

a- La première version

Dans la première version d'IGMP , les routeurs envoient des requêtes d'appartenance de groupe à l'adresse 224.0.0.1 (adresse de diffusion à tous les membres d'un groupe) avec

un TTL=1 sur tous les sous réseaux auxquels ils sont reliés. Quand un hôte reçoit un tel paquet, il répond par un rapport du type "HOST MEMBERSHIP" pour tous les groupes auxquels il appartient. Pour éviter l'avalanche de rapports au niveau du routeur, l'hôte envoie ses rapports après un temps aléatoire qui est tiré à chaque fois qu'il se rend compte qu'un autre hôte a répondu avant lui. Remarque : chaque routeur cherche à savoir quels sont les groupes ayant au moins un membre dans son sous-réseau. Si le routeur ne reçoit pas de rapport pour un groupe particulier après un certain nombre de requêtes, il estime qu'il n'y a plus de membres pour le groupe correspondant et supprime les entrées de ce groupe dans ses tables de routage. Lorsqu'un hôte joint un groupe, il envoie un rapport d'appartenance à ce groupe, avant la requête du routeur, afin de recevoir les paquets adressés à ce groupe.

Dans la première version de IGMP le mécanisme d'élection du routeur qui envoi les messages "query" est fonction du routage multicast et n'appartient pas à IGMP.

b- La seconde version

Dans cette version, le protocole IGMP a été optimisé afin de réduire le temps de latence entre le retrait d'un membre et l'arrêt effectif du flux de paquets lui étant destiné. Cette seconde version a été distribuée avec le paquetage de code du multicast IP (version 3.3 à 3.6). Cette version est compatible avec la première. Elle fournit une procédure d'élection d'un maître, parmi tous les routeurs voisins, seul en charge de la maintenance des informations sur les groupes multicast du sous-réseau. Cette procédure s'est imposée par le fait que jusque là, le routeur qui émettait les requêtes d'appartenance de groupe aux hôtes était désigné par les protocoles de routage multicast, ce qui était source d'inconsistance. C'est le routeur possédant l'adresse IP la plus basse sur le réseau local qui est élu. Lors de l'initialisation, un routeur se considère comme désigné jusqu'à ce qu'il reçoive une requête d'appartenance de groupe d'un routeur d'adresse inférieure à la sienne. Cette version d'IGMP introduit des messages de requêtes adressées à des groupes spécifiques (préférées aux requêtes génériques diffusées en broadcast). IGMP implémente aussi une requête d'abandon de groupe: le message "leave". Si l'hôte qui a répondu en dernier à une requête du routeur décide de quitter un groupe, il envoie un message marquant la fin de

son adhésion au groupe. Le routeur recevant cette requête renvoie alors une requête au groupe spécifié. S'il ne reçoit pas de réponse, après plusieurs requêtes de rapport pour ce groupe, il le supprime des tables de routage de ce sous-réseau.

2.4.3 PIM: Protocol Independant Multicast

On a vu dans la partie précédente que le protocole IGMP réalise la première étape de la transmission du trafic multicast. Tout le reste du travail est réalisé par les protocoles de routage multicast (Multicast Routing Protocol). Pour construire les arbres de diffusion, ces protocoles utilisent plusieurs techniques qu'on peut classer en deux groupes: les techniques basées sur les arbres partagés et celles basées sur la source. PIM est développé dans le cadre du groupe de travail de l'IETF, IDMR (Inter-Domain Multicast Routing). Comme son nom l'indique l'IDMR souhaite développer un protocole de routage entre domaines Internet. PIM doit son nom au fait qu'il ne dépend pas des protocoles de routage unicast sous-jacents. PIM distingue deux modes de fonctionnement pour le protocole : un mode étendu (Sparse Mode) où les membres d'un même groupe peuvent s'étendre sur plusieurs domaines et un mode dense (Dense Mode) où les membres du groupe sont relativement "concentrés". Les protocoles de mode étendu (PIM-SM) sont generalement préférés sur les protocoles de mode dense (PIM-DM).

a- PIM-DM (PIM Dense Mode)

Le mode dense du protocole PIM-DM propage un paquet multipoint, arrivant depuis l'interface menant à l'émetteur, vers toutes les interfaces filles. Ce comportement "je diffuse par défaut" est admissible dans une région à forte densité en membres. PIM-DM est "dirigé par les données", puisque les entrées ne sont créées qu'à l'arrivée du premier message et ne nécessite pas l'adhésion explicite des membres. PIM-DM ne construit pas sa propre table de routage unicast, mais accède simplement aux informations de routes indépendamment du protocole de routage unicast.

b- PIM-SM (PIM Sparse Mode)

PIM-SM permet d'éviter que la présence de membres épars à travers un réseau provoque une innondation du réseau tout entier de paquets multicast. Pour ce faire, le trafic multicast est limité aux seuls routeurs intéressés par le trafic multicast d'un groupe donné.

Les routeurs ayant des membres directement attachés ou des membres en aval doivent rejoindre un arbre de distribution en Sparse-Mode. Un routeur ne faisant pas partie de l'arbre de distribution ne recevra pas des paquets. L'arbre de diffusion multicast partagé est enraciné à un point de rendez-vous (appelé RP) pour tous les membres de groupe. Les sources multicast dans ce domaine envoient les donnés multicast au RP qui les redirige en bas de l'arbre partagé aux récepteurs intéressés dans le domaine. PIM permet aussi de construire un arbre de plus cours chemain relatif à la source et enraciné à celle-ci dans le cas où la source est dans un domaine différent.

PIM-SM implémente le concept de point de rendez-vous (RP) où les récepteurs rencontrent de nouvelles sources. L'initiateur d'un groupe multicast sélectionne un RP primaire et crée une liste ordonnée de RP secondaires (la RP-list). Un seul RP est actif par groupe de multicast. Chaque récepteur souhaitant joindre le groupe envoie un message à son routeur qui à son tour rejoint l'arbre de distribution en envoyant un message au RP. Une source utilise également le RP pour annoncer sa présence et trouver un chemin jusqu'aux membres du groupe.

Dans un domaine PIM, RP est le point de rencontre entre les sources et les récepteurs. Un récepteur adhère tout d'abord à l'arbre partagé du groupe enraciné en RP (figure 2.4).

Un émetteur commence par émettre les données à destination de RP, qui les diffuse sur l'arbre partagé (2.5).

Durant la réception de données à partir de l'arbre partagé, un récepteur peut demander la création d'un arbre de plus courts chemins à une source donnée (2.6).

Dans la table de routage PIM on peut alors trouver des entrées de type (S,G) pour l'arbre enraciné en une source S pour le groupe G, comme on peut trouver des entrées de type (*,G) pour l'arbre partagé de groupe G. Pour chaque entrée on trouve l'adresse unicast de la source ou du RP, l'interface entrante, la liste des interfaces sortantes, et les indicateurs d'état de l'entrée. A la réception d'un paquet adressé au groupe G, un

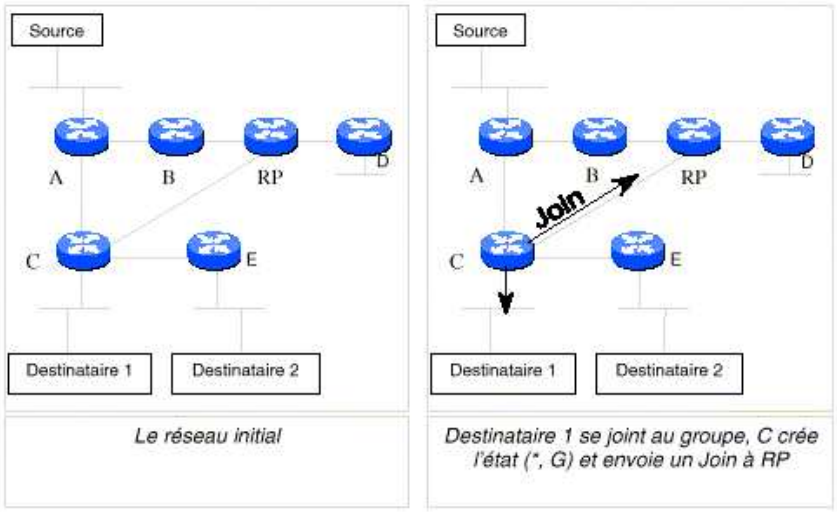

FIG. 2.4 – *Mécanisme d'adhésion d'un membre au groupe dans PIM-SM*

routeur recherche la première concordance dans la table : soit (S,G) existe et l'interface de réception mène à la source, soit (*,G) existe et l'interface de réception mène au RP. Le paquet est alors propagé sur chaque interface sortante , sinon il est abandonné.

Un arbre PIM-SM est robuste car le routeur implémente un mécanisme à rafraîchissement permettant une adaptation aux reconfigurations du routage unicast; le service de diffusion multicast est disponible tant que le service unicast l'est. Quand un membre M décide de rejoindre un groupe G, il avertit son Routeur Désigné DR par le protocole IGMP. DR détermine un RP à partir de l'ensemble des points de rendez-vous RP-Set , il crée alors une entrée (*,G) et émet un message d'adhésion Join.

A la réception du Join, un routeur intermédiaire crée ou met à jour (*,G) (figure 2.6). Dans le cas d'une source qui ne souhaite pas être membre du groupe, le DR crée une entrée (S,G) puis encapsule les données dans un message unicast Register (figure 2.5).

A la réception de ce message, RP le désencapsule et diffuse les données sur l'arbre du groupe. Cette gestion des sources non réceptrices est différente de celle de CBT où une source doit rejoindre l'arbre comme un récepteur normal avant d'émettre. Au cours de la transmission des données, un récepteur peut demander la migration d'un arbre centré vers un arbre par source si le débit d'une source S devient important. Ce DR crée (S,G) et

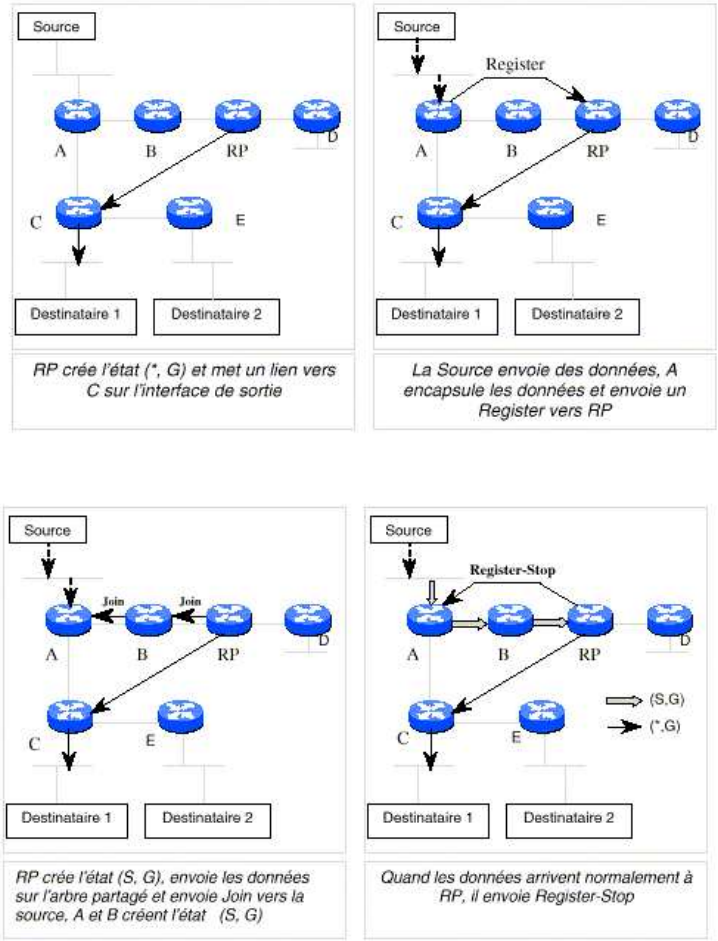

FIG. 2.5 – *Envoi des Donnees par une source dans PIM-SM*

émet un Join à destination de S. Les routeurs intermédiaires créent ou mettent à jour une entrée (S,G). A la réception de données de S, le noeud où l'arbre partagé et l'arbre par source divergent, met à jour son entrée.

2.5 Probèlmes de l'architecture ASM

l'architecture ASM ainsi présentée souffre de plusieurs promblèmes de déploiement dont essensiellement :

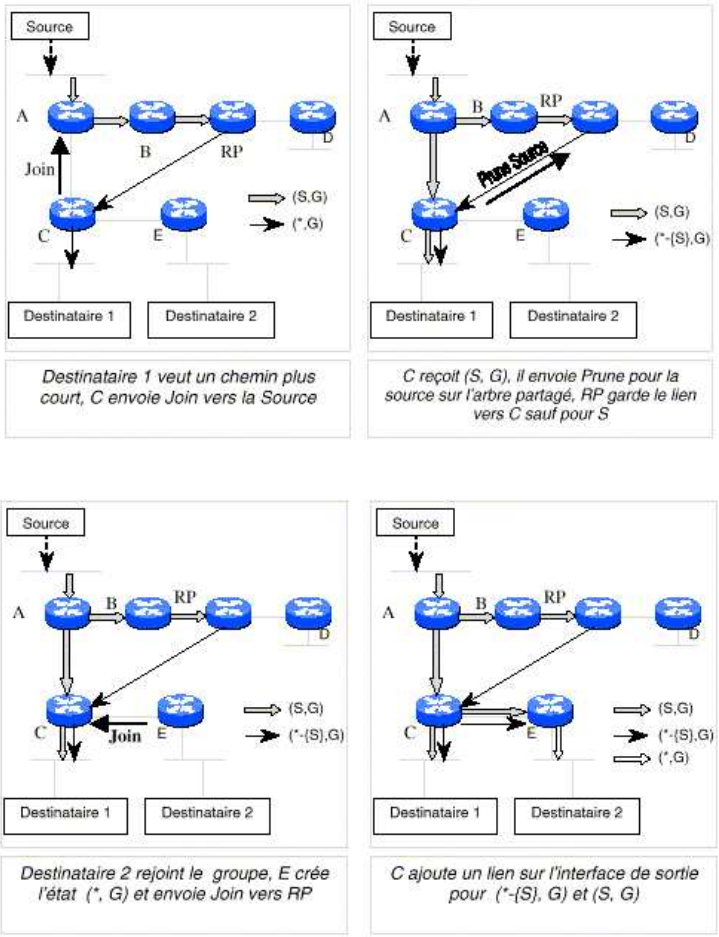

FIG. 2.6 – *Migration vers un arbre de plus cours chemin dans PIM-SM*

2.5.1 Manipulation inefficace des sources bien connues

Dans les cas oú l'adresse de la source est bien connue à l'avance par le récepteur joignant un groupe, et quand l'arbre de plus cours chemain mènant à la source est le mode de redirection préféré, les techniques d'arbre partagé et les mecnismes de MSDP ne sont pas necessaires.

2.5.2 Manque de controle d'accès

Dans le modèle de service ASM, un récepteur ne peut pas indiquer de quelles sources spécifiques il voudrait recevoir le trafic multicast quand il joint un groupe donné. Un récepteur recevra donc toutes les données envoyées à un groupe multicast duquel il est membre par n'importe quelle source.

2.5.3 Allocation d'adresses

L'allocation d'adresse est un des plus grands défis posés au déploiement d'ASM. Cette architecture multicast ne fournit pas de solution deployable pour empécher des collisions d'adresses entre des applications multiples. Ce problme est plus sérieux pour IPv4 que pour IPv6 puisque le nombre d'adresses multicast est plus petit pour IPv4.

2.6 Conclusion

Dans ce chapitre, nous avons essayé de présenter l'architecture Multicast courante ASM (Any source multicast). En particulier, nous avons détaillé les deux mécanismes permettant la communication multicast, à savoir : IGMP et PIM-SM. Le premier pour gérer les groupes dans un réseau local et le second pour assurer le routage multicast. Ainsi, on peut conclure que, dans une architecture ASM, un client peut seulement indiquer son intérêt à un groupe entier de récepteurs et reçoit des données envoyées à ce groupe de n'importe quelle source. Cet aspect, qui correspond au défaut majeur d'ASM, sera résolu dans l'architecture SSM (Source specefic Multicat) qui sera présentée dans le chapitre suivant.

Chapitre 3

Le modèle SSM

3.1 Introduction

Nous avons abouti dans le chapitre précédent à la mise en évidence du principal défaut de l'archirecture ASM. Dans ce modèle, un client peut exprimer son intérêt à recevoir le trafic multicast déstiné à un groupe, mail il est obligé de recevoir les données envoyées à ce groupe de n'importe quelle source. Il n'a pas la possibilité de choisr une source particulière de laquelle il reçoit les données. Des travaux récents dans le domaine du routage multipoint visant à résoudre ce problème ont permis la définition d'un nouveau modèle de service multicast point à multipoint : SSM (Source specefic multicast). Cette approche propose une simplification et une amélioration du service multicast actuel ASM. Elle permet à un hôte de spécifier, en plus du groupe, une liste de sources delaquelle, il désire ou non recevoir le trafic multicast.

Dans ce chapitre, Nous présentons en premier lieu les principeaux apports de l'architecture SSM. Ensuite, nous détaillons le principal composant de cette architecture qui est la troisième version du protocole IGMP.

3.2 Le modèle multicast SSM et ses apports

Le modèle de service SSM définit un canal identifié par une paire (S,G) où S est une adresse de source et G est une adresse de destination SSM. Les recepteurs peuvent recevoir le trafic du groupe G provenant de la source S en s'adhérant au canal (S,G). Les adhésions des membres aux groupes multicast sont gérées pas des protocoles capables d'assurer le filtrage de sources tels que IGMPv3 (pour IPv4) ou MLDv2 (pour IPv6). Ainsi, cette architecture offre les possibilts de filtrage de source qui font qu'un récepteur peut décider de recevoir le trafic d'une source spécifique ou de tous sauf une source spécifique. Seuls les arbres basés et enracinés a la source sont nécessaires pour implémenter ce modèle.

Le modèle de service SSM résoud tous les problèmes de déploiement décrits précédemment :

- SSM offre une solution élégante au problème de contrôle d'accès. Quand un récepteur s'adhére à un canal (S,G), il ne reçoit plus que les données envoyées par la source S. En revanche, dans le modèle ASM, n'importe quel émetteur peut transmettre des données à un groupe multicast.
- SSM définit des canaux dépendant de la source, c'est à dire que le canal (S1, g) est distinct du canal (S2, g), où S1 et S2 sont des adresses de source, et G est une adresse de destination SSM. Ceci évite le problème de l'allocation globale des adresses de destination SSM et rend chaque source indépendamment responsable de résoudre les collisions d'adresse pour les divers canaux qu'elle crée.
- SSM met en oeuvre seulement les arbres de diffusion relatifs à la source. Ceci élimine le besoin en infrastructure pour les arbres partagés. Par consequent, pour la suite de protocoles IGMP/PIM-SM/MSDP, ni les techniques d'arbre partagé et et de point de rendez vous, ni le protocole MSDP sont exigés. Ainsi la complexité de l'infrastructure de routage multicast pour SSM est baissée.
- Les applications point-à-multipoints telles que la télévision sur Internet domineront l'espace des applications multicast sur Internet dans le proche avenir. Le modèle de SSM est parfaitement approprié à de telles applications.

3.3 Le protocole IGMP pour SSM : IGMPv3

Dans la version 3, qui est encore en phase de spécification, le protocole IGMP a été enrichi avec des messages spécifiques aux sources. Un hôte peut spécifier de quelle source il souhaite recevoir le trafic "Group-Source Report", et de quelle source il ne souhaite pas recevoir "Exclusion Group-Source Message", c'est ce que l'on appelle le filtrage. Cette nouvelle considération de la paire Group-Source est très utile quand il existe plusieurs sources pour le même groupe. Remarquons que dans les versions précédentes on a plutôt la notion de Groupe et pas de Groupe/Source. La troisième version permet ainsi une meilleure utilisation de la bande passante.

IGMPv3 est grandement amélioré pour ce qui est des aspects sécuritaires (contre les attaques de type DoS). En effet, il est possible pour un hôte sur un réseau de ne demander à recevoir le trafic multicast pour un groupe que en provenance de certaines sources. Ce filtrage peut se faire de deux façons différentes :

- Mode Inclusion : on accepte les paquets multicast d'un groupe G que si les sources font partie d'un ensemble défini à l'avance et notifié au DR (Designated Router) par le biais d'un message "Version 3 membership report".
- Mode Exclusion : on accepte tous les paquets multicast d'un groupe G sauf ceux provenant de sources dont la liste est transmise par l'hôte au DR par le biais d'un message "Version 3 membership report".

Ce nouveau format de message schématisé dans la figure 3.1 permet de spécifier une liste de "Group Records". Chaque Group Record est décomposé en plusieurs champs qui indiquent au routeur désigné le comportement à adopter pour chaque groupe multicast et qui sont essentiellement les suivants:

1. Multicast adress : adresse de groupe multicast concernée par les changements.
2. Source adress[i] : liste des adresses unicast de sources.
3. Record Type : Ce champ spécifie le type du “group record” envoyé dans le report. Il permet de choisir si le mode est inclusif ou exclusif (current-state record), s'il s'agit d'une mise à jour par ajouts de sources concernées (Source-Liste-Change record) ou

encore s'il s'agit d'un changement d'état (Filter-Mode-change record) d'inclusif vers exclusif ou inversement. On dénombre ainsi 6 types distincts de reports IGMPv3 qui sont les suivantes :

- MODE_IS_INCLUDE (current-state record): indique que le mode de filtrage pour cette interface est "INCLUDE" pour l'adresse multicast spécifiée. Les champs source adress [i] contiennent la liste des sources sur cette interface pour l'adresse multicast indiquée, si elle est non vide.
- MODE_IS_EXCLUDE (current-state record): indique que le mode de filtrage pour cette interface est "EXCLUDE" pour l'adresse multicast spécifiée. Les champs source adress [i] contiennent la liste des sources sur cette interface pour l'adresse multicast indiquée, si elle est non vide.
- CHANGE_TO_INCLUDE_MODE (Filter-Mode-change record): indique que le mode de filtrage de l'interface a changé de EXCLUDE à INCLUDE pour l'adresse multicast indiquée. Les champs source adress [i] contiennent la nouvelle liste des sources sur cette interface pour l'adresse multicast indiquée, si elle est non vide.
- CHANGE_TO_EXCLUDE_MODE (Filter-Mode-change record): indique que le mode de filtrage de l'interface a changé de INCLUDE à EXCLUDE pour l'adresse multicast indiquée. Les champs source adress [i] contiennent la nouvelle liste des sources sur cette interface pour l'adresse multicast indiquée, si elle est non vide.
- ALLOW_NEW_SOURCES (Source-Liste-Change record): indique que les champs source adress [i] dans ce groupe contiennent une liste des sources additionnelles desquelles le sysème souhaite recevoir des données.
- BLOCK_OLD_SOURCES (Source-Liste-Change record): indique que les champs source adress [i] dans ce groupe contiennent une liste des sources desquelles le sysème ne souhaite plus recevoir des données.

Il y a d'autres champs qui ne sont pas décrits ici car ils n'interviennent pas au niveau du principe de fonctionnement du protocole (nombre de Group Records, nombre d'adresses de sources, etc.).

Ce nouveau type "version 3 membership report" n'est pas envoyé de la même façon au DR. Contrairement à IGMPv1 et v2, l'adresse destination de ce paquet n'est pas

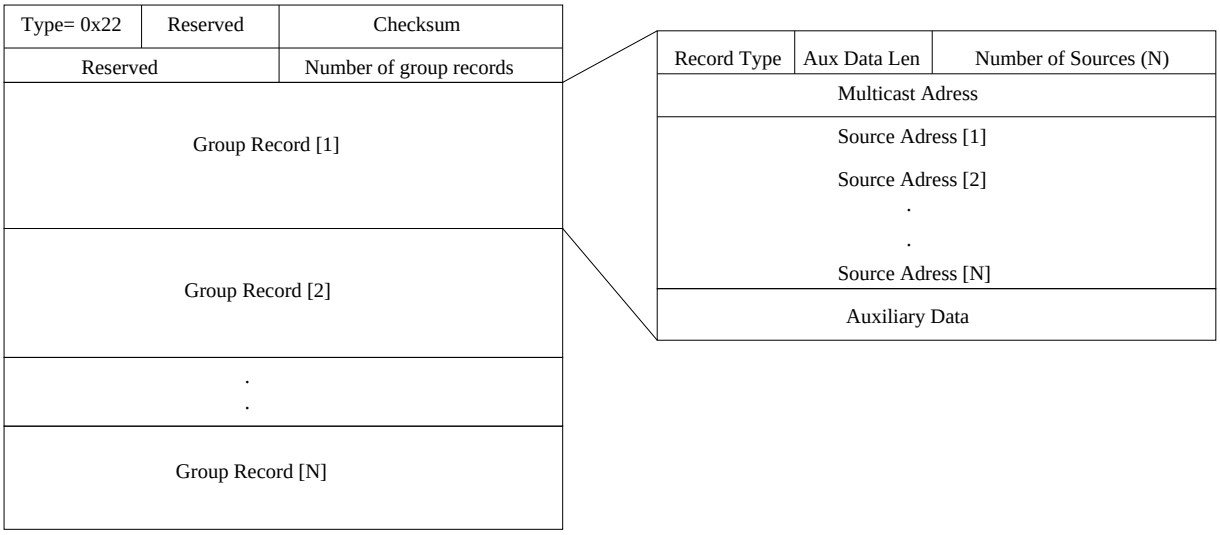

FIG. 3.1 – *Structure d'un group record dans IGMPv3*

l'adresse de groupe concerné. Il s'agit en fait de l'adresse 224.0.0.22 qui désigne les routeurs "IGMPv3 capable".

IGMPv3 couplé avec un protocole comme PIM-SM permet de façon très efficace de joindre un groupe multicast diffusé à partir d'une source bien identifiée. Il s'agit d'ailleurs de la très grande majorité de l'utilisation du multicast à présent et certainement pour le futur (TV ou radio par Internet). Chaque hôte a connaissance des sources à joindre de différentes façons. De manière générale, ceci ne fait pas partie de la problématique d'IGMP. Cependant pour information on peut noter que ces adresses peuvent être acquises de façon applicative, par une page HTML ou encore par SDR.

IGMPv3 nécessite évidemment quelques mises à jour dans les systèmes futurs voulant bénéficier de ces fonctionnalités :

- Mise à jour du noyau système pour la gestion d'IGMPv3,
- Mise à jour des logiciels pour l'utilisation des nouvelles API dédiées à IGMPv3, et qui seules peuvent tirer parti des nouvelles fonctions.

Notons qu'à ce jour IGMPv3 a le soutien de nombreuses entreprises telles Cisco, RealNetworks, Microsoft. Des implémentations d'IGMPv3 sont déjà disponibles pour Linux et

FreeBSD.

Le mécanisme d'abonnement est le même pour IGMPv3 seule la nature des reports est différente.

Avec IGMPv3 le message "leave" a été enlevé. Ainsi, pour que le routeur multicast puisse savoir qu'il n'y a plus de membres pour un groupe sur son réseaux local, il utilise un timer pour chaque groupe qui est mis à jour dès la réception des reports spécifiques à ce groupe. A son expiration il se rend compte qu'il n'y a plus de membres pour ce groupe sur ce réseau local.

3.4 Conclusion

Nous venons ainsi de presenter l'architecture SSM qui sera la plateforme de référence sur laquelle aura lieu nos implémentations ultérieures. Nous avons abouti à dégager les principeaux composant de l'architecture SSM qui sont le protocole de gestion de groupes IGMPv3 et le protocole de routage multicast PIM-SM.

Par ailleurs, dans certaines topologies, il n'est pas toujurs nécessaire d'implanter un protocole de routage multicast dans tous les routeurs. Il suffit de consulter les adhésions aux groupes disponibles dans la table IGMP pour décider d'expédier les paquets multicast sur les differentes interfaces d'un routeur. C'est le principe de fonctionnement du Proxy IGMP.

L'objectif de ce travail est d'étendre la plateforme implémentant l'architecture SSM en lui integrant un Proxy IGMP. Avant de présenter l'implémentation du Proxy IGMP que nous avons réalisé, il convient de détailler l'implementation du protocole IGMPv3-routeur qui a été la squelette sur laquelle a eu lieu l'implémentation du concept de Proxy IGMP.

Chapitre 4

Détails d'implémentation du protocole IGMPv3-routeur

4.1 Introduction

Notre plate-forme de développement est composée de machines sous FreeBSD 4.5. Pour ce système d'exploitation une version IGMPv3 côté hôte [GRA00] est disponible. De plus, la partie routeur d'IGMPv3 a été implémentée au sein du projet RESEDAS [LAH01]. Cette implémentation de IGMPv3 côté routeur est la squelette du proxy IGMP sur laquelle a eu lieu nos impémentations. Afin de bien assimiler l'implémentation du proxy IGMP réalisée, nous avons jugé nécessaire de présenter l'implémentation du coté routeur d'IGMP.

Dans ce chapitre nous détaillons l'implémentation de IGMPv3-routeur tout en donnant une description de ce protocole. Nous mettont l'accent sur les parties les plus imporantes au niveau desquelles nous avons dû interfenir pour réaliser l'implémentation du proxy IGMP.

4.2 Fonctionnalités de IGMPv3-routeur

IGMPv3-routeur permet de réaliser les fonctions suivantes :

1. La réception des différents reports envoyés par les systèmes voisins. Pour indiquer

son état un hôte supportant IGMPv3 peut envoyer deux types de reports, et les actions prises par le routeur diffèrent selon le type de report reçu.

- Réception d'un "current-state" report : à la réception de ce type de report le routeur met à jour le timer du groupe ainsi que ceux des sources.
- Réception d'un "Filter-Mode-change" ou "Source-List-change" report : Ils s'agissent d'un "state-change" report. Quand l'état global IGMP d'un groupe change dans un système (hôte), alors ce dernier envoie un report de type "Filter-Mode-Change" ou "Source-List-Change". A la réception de ce report, en plus de la mise a jour de timer du groupe et des sources, le routeur doit envoyer des "query" pour solliciter l'ensemble des sources contenu dans le report, et ceci pour vérifier s'il y a des membres qui désirent ces sources ou non.

2. L'envoi des "query" IGMP : un routeur multicast doit envoyer des "General-query" périodiquement vers les systèmes qui lui sont directement attachés pour connaître leur état IGMP. Les systèmes doivent répondre par des "reports". Les différents types de "query" envoyés par un routeur sont :

   - " Group-Specific-Query " : utilisé pour vérifier si un système désire la réception pour un groupe particulier ou pour reconstruire l'état IGMP de ce groupe. Ce message est envoyé en réponse à "state-change record" indiquant qu'un système a quitté le groupe.
   - " Group-Source-Specific-Query" : utilisé pour vérifier si un système désire recevoir du trafic pour un ensemble de sources et un groupe spécifique. Ce type de "query" est envoyé en réponse à "state-change record" indiquant un changement d'état d'un membre du groupe. Ce message est jamais envoyé en réponse à "current-state record".

3. Le maintien et la mise à jour d'un état IGMP pour chaque groupe sur chaque réseau local.
   Un état IGMP est défini comme ceci :

   (Adresse multicast, Timer de groupe, Mode : INCLUDE/EXCLUDE, Les sources)

- "Adresse multicast" : est une adresse de classe D.
- "Timer de groupe" : est utile seulement si le mode est EXCLUDE. Il représente le temps nécessaire pour que le mode de filtrage change en INCLUDE quand il expire. Ce timer est mis à jour suivant les reports reçus pour ce groupe. Quand le mode de filtrage pour un groupe est EXCLUDE et que son timer expire, alors aucun membre de ce groupe est en mode EXCLUDE. Dans ce cas, le routeur change le mode de filtrage de ce groupe en INCLUDE.
- "Mode" : indique le mode de filtrage d'un groupe. Il peut être INCLUDE ou EXCLUDE. Les messages IGMPv3 "reports" reçus pour ce groupe ainsi que son timer permettent de transiter d'un mode à un autre.
- Chaque champ source est de la forme suivante :

  (Adresse de la source, Timer de la source)

  L'adresse de la source est une adresse unicast. Si toutes les sources sont désirées par un groupe alors ce champ est vide et le mode de groupe sera EXCLUDE, ceci est équivalent à joindre le groupe en IGMPv1 ou IGMPv2. Le timer de source sert à indiquer au protocole de routage multicast s'il y a des membres qui souhaite recevoir ou non du trafic de cette source. Ce timer est mis à jour par les rapports/reports reçus pour un groupe. Pour le mode de filtrage EXCLUDE il a deux types de sources:

  - Les sources avec un timer strictement positif et qui présentent un conflit entre les membres d'un groupe (certains veulent les inclure et d'autres non).
  - Les sources ayant un timer nul et qui ne sont pas souhaitées par les membres du groupe.

  Pour le mode de filtrage INCLUDE, il existe un seul type de sources : celles qui intéressent tous les membres du groupe.

4. La réception des "query" : à la réception des " query " le routeur met à jour les timers de sources et de groupes selon le type de "query ", si le bit " Clear Suppress

Router-Side Processing " est mis à 0. Ce bit sert pour la synchronisation de la mise à jour des timers entre les routeurs multicast, et le routeur responsable de l'envoi des "query". Un mécanisme pour l'élection des " querier " est utilisé, en choisissant le routeur de plus faible adresse IP comme "querier ".

5. L'interoperabilité avec les autres versions de IGMP : les hôtes et les routeurs supportant IGMPv3 doivent assurer la compatibilité avec les autres versions (IGMPv1, IGMPv2). Pour chaque état IGMP d'un groupe le routeur doit définir un mode de compatibilité. Ce mode détermine la version de IGMP utilisée pour ce groupe. Différents timers sont prévus pour passer d'un mode à un autre, par défaut le mode est IGMPv3. Donc le membre de groupe de plus faible version de IGMP va définir le mode de compatibilité du groupe.

4.3 Détails de l'implémentation

Dans cette section nous présentons les détails de l'implémentation de IGMPv3-routeur.

4.3.1 Architecture générale de l'implémentation

L'implémentation de IGMPv3-routeur maintient une architecture modulaire. La figure 4.1 montre l'architecture générale de cette implémentation :

Quand le noyau reçoit un paquet IP, il le passe à la routine IP-intr (fichier "ip_input.c"). Cette routine regarde l'adresse de destination et le numéro de protocole et l'envoie à la routine adéquate. Dans le cas d'un message IGMP, une copie du paquet est envoyée à la routine IGMP_input, pour le traiter par la partie hôte de IGMP (dans le cas où la partie hôte est installée). Une autre copie est délivrée au socket IGMP du démon "igmppd" et à tous les autres sockets qui sont à l'écoute de ce type de message.

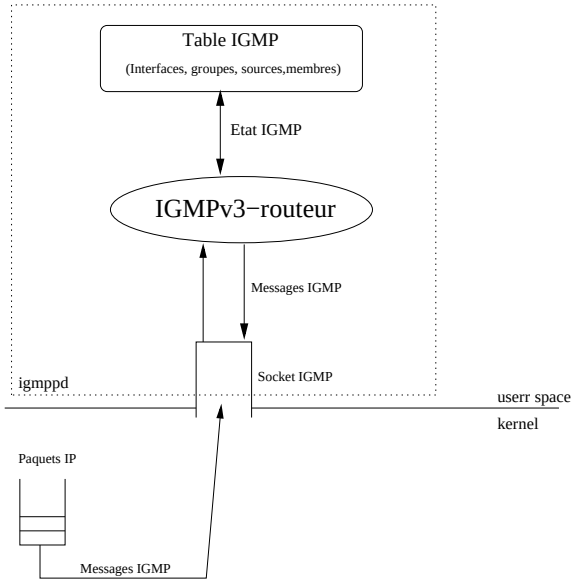

FIG. 4.1 – *Architecture générale de l'implémentation du protocole IGMPv3-routeur*

4.3.2 Description des fonctions

Voici une description des principales routines du démon "igmppd" correspondant au protocole IGMPv3-routeur:

- Igmprt_input : un thread est associé à cette fonction, il permet d'écouter les messages IGMP sur le socket raw et dès l'arrivée d'un message il le passera à la routine adéquate pour le traiter.
- Igmprt_timer : un thread est associé à cette fonction, il permet de mettre à jour le timer de chaque groupe, de chaque source et d'envoyer les " query " planifiés sur chaque interface.
- Igmprt_timer_source : permet de mettre à jour le timer de chaque source.
- Igmprt_timer_group : met à jour le timer de chaque groupe et effectue les traitements adéquats s'il expire.
- Receive_membership_query : il traite la réception d'un query. Dans ce cas, le bit " Supress Router-Side Processing " est examiné. S'il est mis à 0 alors s'il s'agit d'un

query de type "group specific" et le timer de groupe est mis à jour, sinon il s'agit d'un query de type " group and source specifc query " et le timer de chaque source du groupe en question est mis à jour.

- Igmp_interface_membership_report_v12 : C'est la fonction qui traite un report de type IGMPv1/v2. Un report de ce type est considéré comme un " IS_EX " dans la logique de IGMPv3.
- Igmp_interface_membership_report_v3 : C'est la fonction qui traite un report de type IGMPv3. Selon le type de report, cette fonction appelle la routine adéquate.
- Igmprt_membership_query : cette fonction envoie des query, la version de l'interface est examinée pour déterminer le type de query (IGMPv1/v2, IGMPv3).
- Send_sh_query : cette fonction permet d'envoyer périodiquement, les query planifiés par le routeur sur chaque interface.

La figure 4.2représente les différentes relations entre les fonctions.

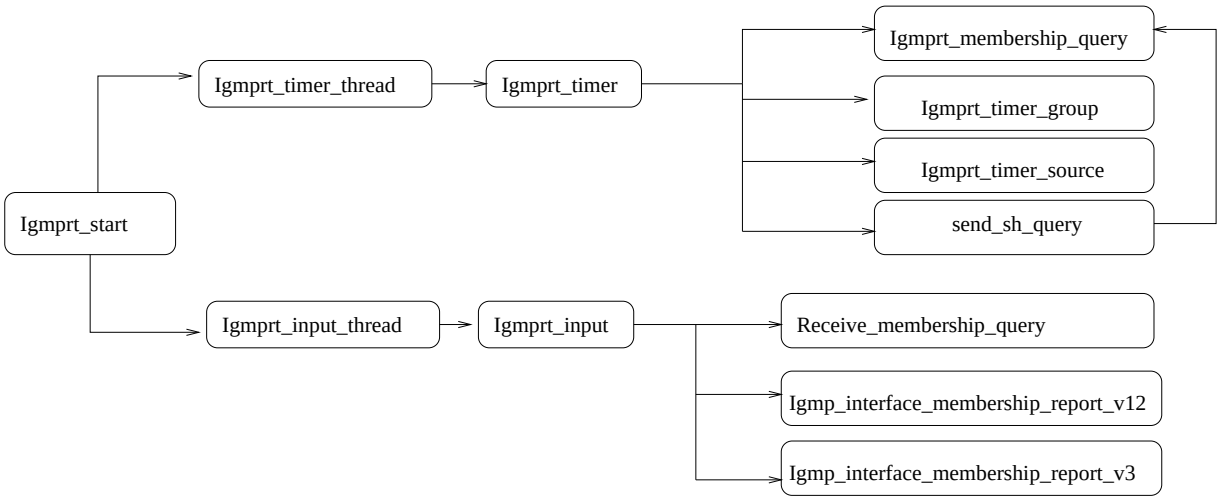

FIG. 4.2 – *Les relations entre les différents fonctions de l'implémentation de IGMPv3-routeur*

4.3.3 Diagramme de flow de données

A la réception d'un message IGMP, le démon "igmppd" va identifier son type et l'envoie à la routine adéquate pour le traiter et mettre à jour la table IGMP. En même temps un processus (thread) est associé à un timer qui à chaque intervalle de temps appelle les routines de gestion des timers de groupes, des timers de sources, et la routine d'envoi des sollicitations planifiées par le routeur. La figure 4.3 représente le diagramme de flow de données de notre démon.

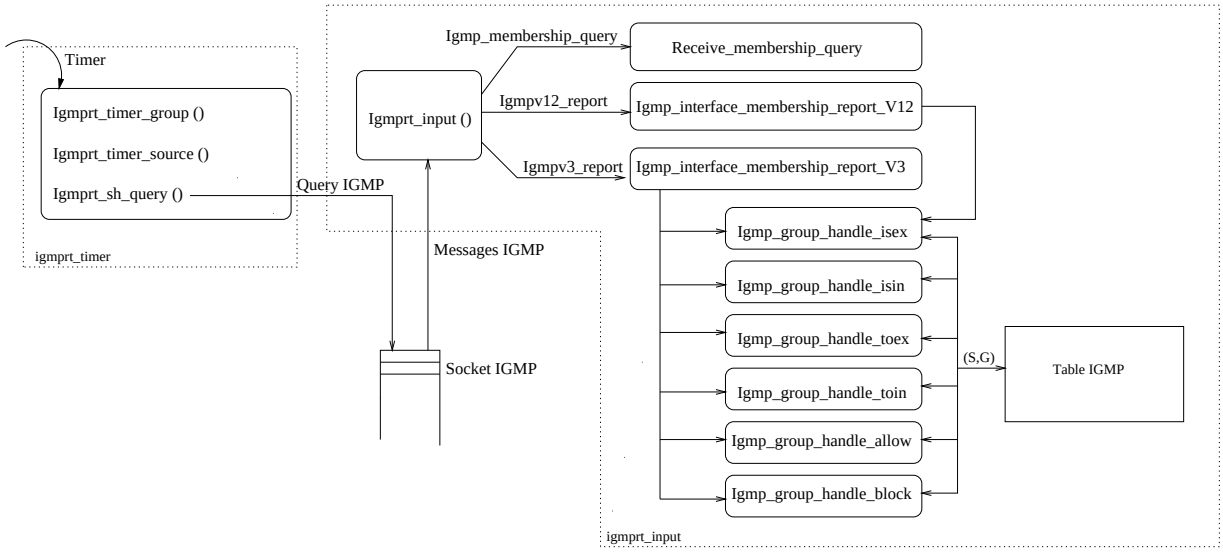

FIG. 4.3 – *Diagramme de flow de données de l'implémentation de IGMPv3-routeur*

4.3.4 Structures de données

La figure 4.4 définit les structures de données importantes utilisées dans cette implémentation, ainsi que les relations entrent elles.

Les structures de données principales de l'implémentation sont :

- La structure " igmp_router_t " : c'est la structure principale, elle englobe toutes les informations concernant un routeur multicast. Elle permet de mémoriser les données de ce routeur : descripteur des threads (timer et entrée) et les interfaces.

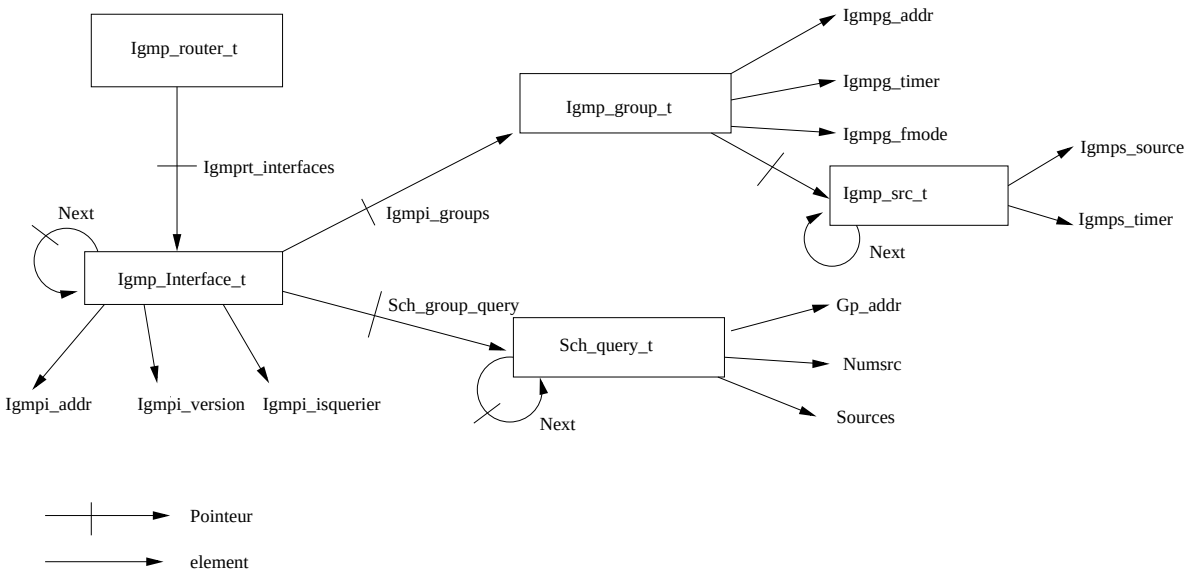

FIG. 4.4 – *Les Structures de données de l'implémentation de IGMPv3-routeur*

- La structure " igmp_interface_t ": cette structure contient les informations concernant une interface. Chaque interface contient un descripteur de socket, un ensemble de groupes, la version d'IGMP et l'ensemble de " query " planifiés par le routeur.
- La structure " igmp_group _t ": cette structure contient les informations concernant un groupe. Chaque groupe possède un timer pour assurer le passage d'un mode de filtrage à un autre, un ensemble des sources demandés par les membres de ce groupe, un mode de filtrage (INCLUDE/EXCLUDE), et l'ensemble des membres de groupe sur chaque interface, qui sera utile pour l'authentification de ces membres dans le cadre de l'implémentation du protocole Baal.
- La structure " igmp_src_t ": Cette structure contient les informations concernant une source demandée par les membres d'un groupe. Chaque source possède un timer qui indique son état.
- La structure " igmpv3q_t ": C'est la structure d'une " query " IGMPv3 comme elle était définie dans l'Internet draft de IGMPv3.
- La structure " igmpr_t ": C'est la structure d'un report IGMPv1/IGMPv2, qui

est gardée dans notre implémentation pour assurer la compatibilité avec les autres modes.

- La structure " igmp_grouprec_t ": C'est la structure d'un enregistrement d'un groupe "group record" dans un report IGMPv3.
- La structure " igmp_report_t ": C'est la structure d'un report IGMPv3.

4.4 Conclusion

L'implémentation de IGMPv3 côté routeur ainsi présentée peut être considérée comme un squelette du proxy IGMP, qui est en cours de spécification par l'IETF. Un proxy IGMP a pour rôle de remplacer les routeurs PIM-SM dans un domaine. Il permet aussi d'assurer la compatibilité entre les routeurs PIM-SM supportant seulement IGMPv2 et les hôtes utilisant IGMPv3 pour envoyer leurs demandes d'abonnement.

Dans le chapitre suivant nous détaillerons l'architecture et le principe de fonctionnement du proxy IGMP. En suite, nous présenterons des détails sur son implémentation.

Chapitre 5

Spécification des composants et des fonctionnalités du Proxy IGMP

5.1 Introduction

Avant d'entamer la déscription de l'impémentation du proxy IGMP, il est nécessaire de spécifier les différents composants ainsi que ses principales fonctinnalités. Dans ce chapitre, nous présentons l'architecture dans laquelle sera intégré le proxy IGMP. A partir de cette architecture, nous avons identifié les différents composants du proxy ainsi que les fonctions qu'il doit assurer. Ces fonctions consistent simplement en la réception des paquets multicast et puis leur envoi (forwarding) aux membres adhérés qui désirent receoir ces paquets. Ainsi, nous détaillons dans ce qui suit les mécanismes qui assurent les dites fonctions requises par le proxy.

5.2 Composants du proxy IGMP

L'architecture du réseau supportant le service multicast, étant assimilée à un arbre, le proxy IGMP comportera une interface upstream unique et plusieurs interfaces downstream. L'interface upstream est celle qui mène à la racine de l'arbre. Elle est directement liée à un routeur multicast supportant PIM-SM ou à un autre proxy IGMP. Cependant,

une interface downstream sera liée à un réseau LAN contenant les membres des groupes multicast. Cette architecture, représentée dans la figure 5.1 correspond à une architecture SSM intégrant un Proxy IGMP.

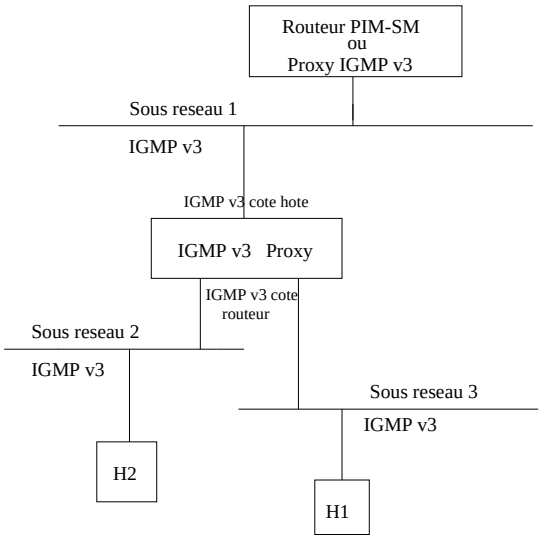

FIG. 5.1 – *Architecture SSM utilisant un Proxy IGMPv3*

Comme représenté dans la figure 5.1, le proxy IGMP doit assurer la partie routeur du protocole IGMP sur ces interfaces downstream et la partie hôte de ce protocole sur son interface upstreaum. Le proxy maintient et met à jour la table IGMP. Rappelons que cette table consiste en une liste contenant pour chaque interface downstream les adhésions aux différents canaux (S,G) sous la forme d'un enregistrement (adresse de groupe , mode de filtrage , timer de groupe , liste de sources) où chaque élément de la liste de sources est un enregistrement (adresse de source , timer de source) . La table IGMP contient donc toutes les informations nécessaires pour que le proxy puisse prendre la décision de rediriger les paquets multicast sur ces différentes interfaces. C'est en se référant aux informations disponibles dans le table IGMP que proxy décidera d'envoyer ou non les paquets multicast sur ses diffféntes interfaces.

5.3 Fonctionnalités du proxy IGMP

Comme nous l'avons déjà énoncé, le proxy doit gérer la réception des paquets multicast afin de les rediriger aux membres qui les demandent.

5.3.1 Gestion de la réception des paquets multicast

Le proxy reçoit sur ces interfaces downstream les reports IGMP venant des systèmes voisins. A chaque reception d'un report, le proxy met à jour sa table IGMP. On suppose que les reports supportés par le proxy sont tous des reports IGMPv3. Les reports IGMPv1 et v2 sont transformés en v3. A chaque modification de la table IGMP (ajout d'une nouvelle entrée, suppression ou modification d'une ancienne entrée), le proxy envoie des reports sur son interface upstream pour notifier le routour voisin du changement de sa table IGMP. Ces reports sont envoyés pas la prtie hote du protocole IGMPv3 et premttent au proxy de recevoir le trafic multicast pour les groupes demandés. Le Proxy se comporte ainsi comme un hôte qui, en recevant une demande d'adésion sur l'une des ses interfaces downstream, envoie à son tour une demande d'adhésion pour le même groupe sur l'interface upstream et ainsi recevoir le trafic pour ce groupe et le transmettre par la suite. La figure 5.2 montre comment un proxy doit joindre sur son interface upstream tous les groupes pour lesquels il y des membres sur ces differentes interfaces downstream.

D'autre part, le Proxy doit reporter sur son interface upstream toutes les demandes de permission ou de blockage du trafic d'une source reçues sur les interfaces downstream. Il effectue ainsi le filtrage sur les sources en évitant de recevoir le trafic multicast des sources non désirées sur toutes les interfaces downstream. Le proxy effectue, grace a la prtie hote du protocole IGMPv3 la concaténation des états IGMP de toutes ses interfaces downstream et envoie sur l'interface upstream l'état résultant (figure 5.3). Ceci permettra au proxy de recevoir tous le trafic multicast de toutes les sources demandées.

Dans la figure 5.3, nous avons supposé en premier lieu que deux membres H1 et H2 sur les interface downstream désirent joindre un groupe multicast G. Le premier (H1) veut recevoir le trafic de le source S (Include(S)) tandis que le second (H2) ne le désire pas (Exclude(S)). La concatenation de ces deux états sera Include (S). Le proxy envoie sur

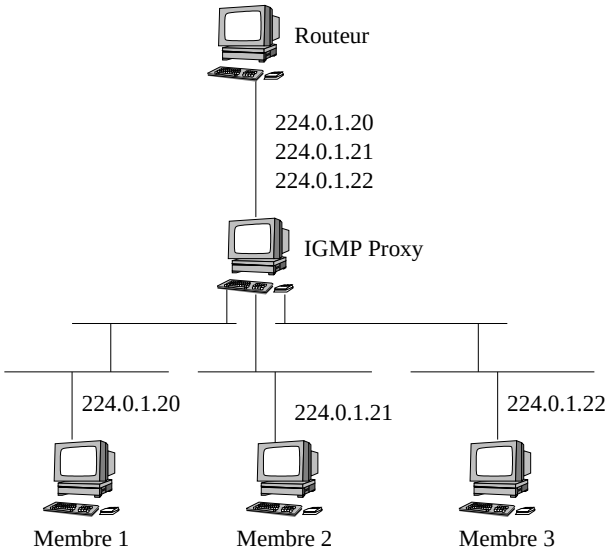

FIG. 5.2 – *Fonctionnement d'un proxy IGMP*

l'interface upstream un report pour permettre les paquets multicast venant de la source S au groupe G. Ainsi, il recevra le trafic correspondant au canal (S,G) pour l'envoyer par la suite a H1 et pas a H2.

Supposons maintenant que les deux membres H1 et H2 ne désirent pas recevoir le trafic de la source S (Exclude(S)). L'état résultant sera Exclude (S) et le proxy envoie un report sur son interface upstream pour bloquer le trafic multicast venant de la source S et destiné au groupe G. Il évite ainsi de recevoir ces paquets.

5.3.2 Mécanisme de redirection des paquets multicast

Le proxy IGMP, étant dépourvu de protocole de routage multicast, il est nécessaire de compléter l'implementation de IGMP-routeur par les instructions nécessaires au forwarding des paquets multicast. C'est la deuxieme fonctionnalité que doit assurer le proxy IGMP.

Le proxy doit transmettre les paquets reçus sur son interface upstream à chaque interface downstream en se basant sur les adhésions des interfaces downstream aux différents canaux multicast. D'autre part, le proxy doit transmettre les paquets reçus sur n'importe

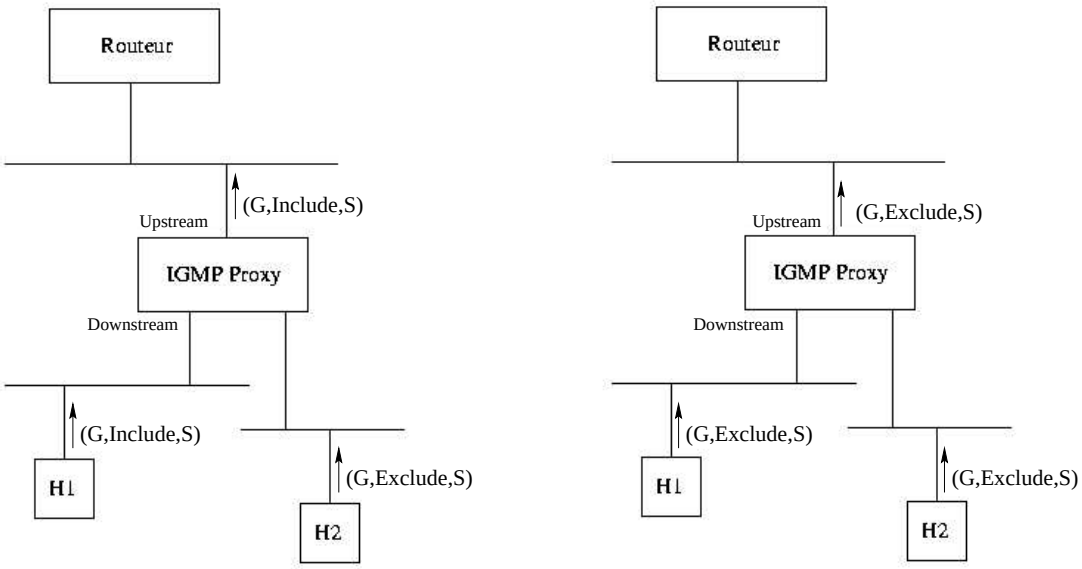

FIG. 5.3 – *Filtrage sur les sources effectué pas le proxy IGMP*

quelle interface downstream à l'interface upstream et à chaque interface downstream autre que l'interface entrante en se basant sur les adhésions des interfaces downstream.

Quand le noyau reçoit un datagramme d'une source destinée à un groupe particulier, une décision de rediriger le datagramme sur chacune des interfaces downstream doit être prise par le démon. Le proxy utilise une cache implémentée au niveau noyau du système d'exploitation et appelée "multicast forwarding cache" (MFC) afin de ne pas prendre la décision de forwarding pour chaque paquet. C'est dans cette cache que les informations de redirection des paquets multicast sont disponibles. La redirection des paquets multicast, étant effectuée par le noyau, le démon igmp-proxy doit donner ces consignes de redirection au noyau en mettant à jour cette cache à chaque changement d'une partie quelconque de information utilisée pour la construire.

Le demon igmp-proxy communique donc avec le noyau pour lui envoyer les messages d'initialisation, d'ajout, de suppression ou de mise à jour de la MFC et de la liste des interfaces virtuelles. Cette liste contient des informations sur les différentes interfaces du proxy, elle est utilisée pas le noyau lors de la redirection des paquets multicast.

L'implémentation du protocole IGMP proxy aura comme squelette celle du protocole IGMPv3-routeur. Certaines modifications sont nécessaires pour introduire les différentes fonctionnalités requises par le proxy.

5.4 Conclusion

Ainsi, on peut recapituler les principales fonctions que doit effectuer un proxy IGMP pour assurer le routage multicast. Il doit, en premier lieu, gérer la réception de paquets multicast. Ceci se fait en s'adhérant aux groupes nécessaires et en blockant ou permettant le trafic des sources selon les demandes des membres sur ces interfaces downstraem. Il doit par la suite rediriger les paquets sur ces differentes interfaces en fonction des adhésions des différents membres. Dans le chapitre qui suit nous expliquons comment nous avons implémenté ces différents concepts.

Chapitre 6

Détails d'implémentation du Proxy IGMP

6.1 Introduction

Aprés avoir spécifié les fonctionnalités du proxy IGMP, nous présentons dans ce chapitre les détails d'implémentation du proxy. Ainsi nous détaillons successivement l'implémentation de la partie permettant le réception des paquets multicast puis celle permettant leur redirection.

6.2 Réception des paquets multicast

Nous avons noté dans ce qui précède que le proxy IGMP assure les protocoles IGMP hôte sur son interface upstream et IGMP routeur sur ses interfaces downstream. Le protocole IGMP hôte, étant implémenté au niveau noyau, il n'y a pas de démon correspondant à ce protocole. Un moyen de communication entre ces deux protocoles est nécessaire pour permettre au proxy d'envoyer les messages nécessaires au routeur upstream à chauqe réception d'un report sur son interface downstream. Le démon IGMP routeur se comportera comme un membre sur l'interface upstream pour chacun des groupes auxquels les membres des interfaces downstream sont adhérés.

Les appels système "setsockopt" et “ioctl” utilsés avec des options disponibles dans l'API d'IGMPv3 permettent de notifier le noyau, en l'occurence le protcole IGMP hôte, du chagement de la table IGMP. ils lui demandent par example de jondre un groupe, bloquer ou permettres le trafic d'une source etc. Ce dernier s'occupera de l'envoi des reports nécessaires sur l'interface upstream. Dans ce qui suit, nous allons présenter les différentes utilisations des appels système "setsockopt" et ”ioctl” auxquells nous avons eu recours selon le type de report reçu.

6.2.1 Joindre un groupe sur l'interface upstream

La fonction “igmp_interface_group_add” fait partie de l'implémentation d'IGMPv3-routeur. Elle permet d'ajouter un groupe à la liste des groupes d'une interface dès qu'un premier membre sur le réseau connecté à cette interface désire joindre ce groupe. Nous avons ajouté à cette fonction les instructions nécessaires pour que le demon igmp-proxy demande à la partie hôte d'IGMP de joindre le groupe multicast en question sur l'interface upstream. il s'agit d'utiliser l'option socket IP_ADD_MEMBERSHIP avec l'appel système ”setsockopt” comme suit :

```
if (setsockopt(router->igmprt_up_socket, IPPROTO_IP, IP_ADD_MEMBERSHIP,
(void *) &mreq, sizeof(mreq)) <0) {
   perror("setsockopt - IP_ADD_MEMBERSHIP");
   exit(1);
}
```

où mreq est une structure de type ip_mreq, définie dans le fichier /usr/include/netinet/in.h, qui est décrite comme suit :

```
struct ip_mreq {
       struct in_addr imr_multiaddr;   /* adresse multicast du groupe */
       struct in_addr imr_interface;   /* adresse IP de l'interface */
};
```

L'initialisation de la structure mreq est la suivante :

```
mreq.imr_multiaddr.s_addr=groupaddr.s_addr;     /*adresse du groupe */
mreq.imr_interface.s_addr=upstream;  /*adresse de l'interface upstream*/
```

6.2.2 Bloquer ou permettre de trafic d'un ensemble de sources

A chaque réception d'un report, le demon IGMP-Proxy doit, en plus des traitement usuels effectués par le protocole IGMPv3-routeur, reporter sur son interface upstream toutes des demandes de permission ou de blockage du traffic d'une source. C'est le filtrage sur les sources duquel le proxy doit être responsable.

Comme mentionné avant, le protocole IGMPv3-routeur distingue 6 types de groupes records qui peuvent être envoyés dans les reports IGMPv3. A chaque report reçu, et selon son type, un changement dans la table IGMP doit être effectué par le démon igmp-routeur. Ces changements dépondent aussi du mode de filtrage initial du groupe dans la table IGMP. Le démon IGMP-Proxy doit envoyer au noyau, en l'occurence le protocole IGMP-hôte, l'état IGMP final résultant de chaque report. Le protocole IGMP-hôte se contentera d'envoyer sur l'interfece upstream les reports nécessaires. Il effectue ainsi le filtrage sur les sources en fonction des demandes des membres sur les interfaces downstream.

Les figures 6.1, 6.2 et 6.3 présentent les differents changements affectés à la table IGMP à chaque réception d'un report selon son type et le mode de filtrage initial. Ces figures, sous forme de diagrammes d'état, présentent les actions que le proxy doit effectuer pour gérer la réception des paquets multicast sur son interface upstream afin d'effectuer le filtrage sur les sources en fonction des demandes des différents membres. Ces actions sont tous implémentés grâce à l'appel système ioctl() qui communique au noyau une adresse de groupe, un mode de filtrage et une liste de sources sur une interface donnée pourque celui-ci envoie les reports IGMPv3 nécessaires.

Les notations suivantes sont adoptées :

- IS_IN (X) : désigne un report MODE_IS_INCLUDE avec X la liste de sources.

- IS_EX (X) : désigne un report MODE_IS_EXCLUDE avec X la liste de sources.
- TO_IN (X) :désigne un report CHANGE_TO_INCLUDE_MODE avec X la liste de sources.
- TO_EX (X) : désigne un report CHANGE_TO_EXCLUDE_MODE avec X la liste de sources.
- ALLOW (X) : désigne un report ALLOW_NEW_SOURCES avec X la liste de sources.
- BLOCK (X) : désigne un report BLOCK_OLD_SOURCES avec X la liste de sources.
- (X , Y) : désigne une liste de sources où X contien les sources dont le timer est strictement positif et Y contient les sources dont le timer est nul.
- A+B : désigne la réunion des listes de sources A et B.
- A*B : désigne l'intersection des listes de sources A et B.
- A-B : désigne la liste A delaquelle on a supprimé les éléments de B.

Ces diagrammes résument les actions de filtrage de sources prises par le proxy à chaque réception d'un report. En d'autes termes, Ces figures presentent le mode de filtrage et la liste de sources que nous devons passer en parametre de l'appel systeme "ioctl" pour que le protocole IGMPv3-hote envoie les reports necessaires sur l'interface upstream.

L'appel système "ioctl()" que nous utlisons appelle l'option "SIOCSIPMSFILTER" et se fait comme suit :

```
if (ioctl(router->igmprt_up_socket, SIOCSIPMSFILTER, imsfp) != 0){
   perror("ioctl - SIOCSIPMSFILTER");
}
```

où imsfp est un pointeur sur une structure de type ip_msfilter, définie dans le fichier /usr/include/netinet/in.h, et décrite comme suit :

```
struct ip_msfilter {
   struct in_addr imsf_multiaddr;   /* addresse du groupe multicast */
```

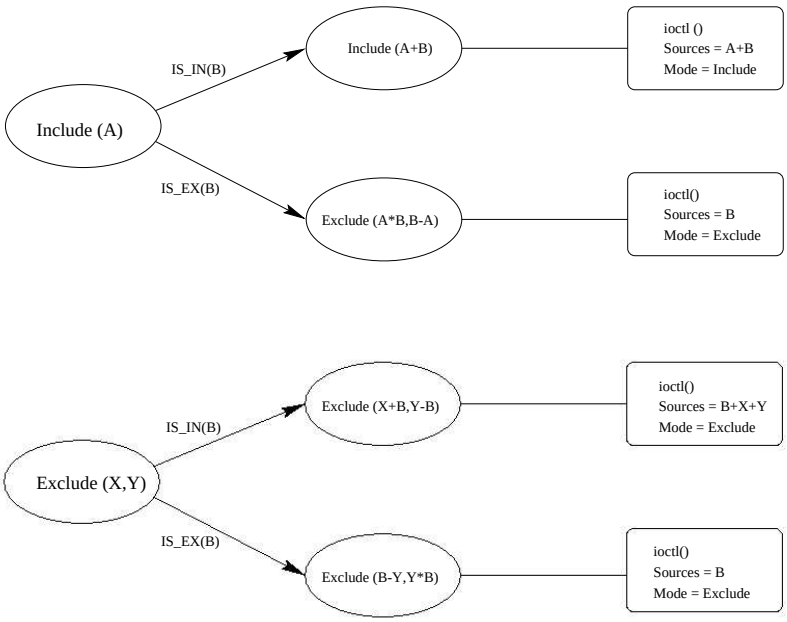

FIG. 6.1 – *Diagramme d'état d'un "current_state" report pour le proxy IGMP*

```
    struct in_addr imsf_interface;   /* addresse IP de l'interface */
    uint32_t imsf_fmode;          /* mode de filtrage */
    uint32_t imsf_numsrc;         /* nombre de sources dans la liste */
    struct in_addr imsf_slist[1];     /* début de la liste de sources */
};
```

6.2.3 Quitter un groupe sur l'interface upstream

Pour optimiser au mieux le fonctionnement du proxy IGMP, il est nécessaire d'arréter la réception des paquets multicast sur l'interface upstream lorsque tous les abonnements à un groupe donné sur toutes les interfaces downstream sont arrétés. L'option IP_DROP_MEMBERSHIP de l'appel système "setsockopt" permet de désinscrire une interface d'un groupe donné. Dans notre cas, il s'agit de supprimer le groupe en question de l'entrée correspondante à l'interface upstream dans la table IGMP a chaque besoin. Le recours à cet appet se fait à chaque désabonnement d'un groupe d'une interface downstream donnée après vérification qu'aucune autre interface downstream ne présente des

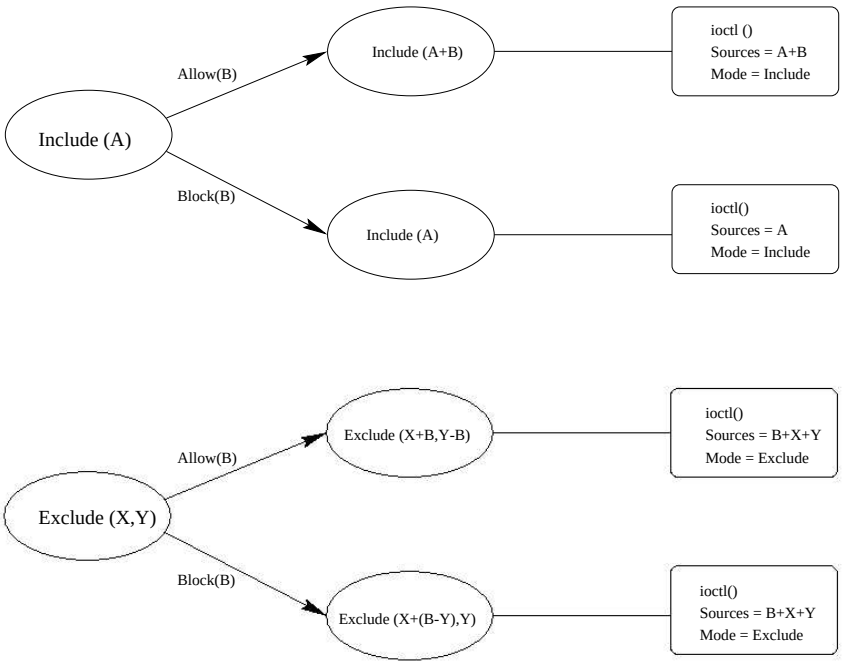

FIG. 6.2 – *Diagramme d'état d'un "source_list_change" report pour le proxy IGMP*

abonnements pour ce groupe. l'implementation de cette partie se fait comme suit :

```
if (setsockopt(router->igmprt_up_socket, IPPROTO_IP, IP_DROP_MEMBERSHIP,
(void *) &mreq, sizeof(mreq)) <
  perror("setsockopt - IP_DROP_MEMBERSHIP");
  exit(1);
}
```

ou "mreq" est une structure du type ip_mreq.

6.3 Forwarding des paquets multicast

L'implementation du mécanisme de redirection des paquets multicast nécessite plusieurs opérations. En premier lieu, il faut initialiser le processus du routage multicast puis

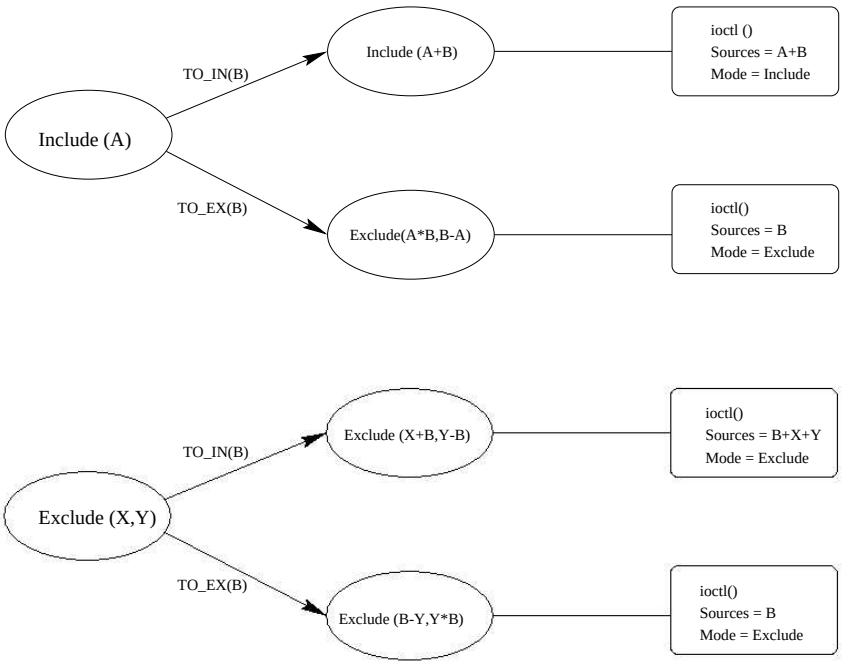

FIG. 6.3 – *Diagramme d'état d'un "filter_mode_change" report pour le proxy IGMP*

installer au niveau noyau la liste des interfaces virtuelles. Il faut aussi mettre a jour la MFC à chaque changement de la table IGMP. C'est dans cette cache que sera enregistré les consignes de redirection au noyau. Par ailleurs, à l'arrêt du routeur, il faut désactiver le processus de routage multicast.

6.3.1 Initialisation du routage multicast

Pour permettre la redirection des paquets multicast par le noyau, il est nécessaire d'initialiser le processus de routage multicast. Cette opération se fait en appelant la fonction du noyau ip_mrouter_init() qui existe dans le fichier /usr/src/sys/netinet/ip_mroute.c .

Pour appeler la fonction ip_mrouter_init(), nous avons eu recours à l'appel système setsockopt avec l'option MRT_INIT comme suit :

```
int v = 1;
if (setsockopt(socket, IPPROTO_IP, MRT_INIT, (char *)&v, sizeof(int)) <0)
    perror("setsockopt - MRT_INIT");
```

6.3.2 Installation de la liste des interface virtuelles

Le proxy IGMP accepte les datagrammes multicast entrants, les duplique et retransmet les copies sur une ou plusieurs interfaces. Par ce biais, les datagrammes sont retransmis aux autres routeurs ou aux différents membres de l'inter-réseau.

En général, une interface de sortie peut être soit une interface physique ou un tunnel multicast. Les tunnels permettent à deux routeurs multicast d'échanger des datagrammes multicast par l'intermédiaure de routeurs ne suportant pas le multicast. Cependant, une interface physique est connectée directement soit à un résaeu local contenant les membres événtuels des groupes multicast, soit à un autre proxy IGMP.

Aussi bien pour les interfaces physiques que pour les interface tunnel, le noyau gère une table d'interfaces virtuelles contenant des informations servant uniquement au multicasting. Chque interface virtuelle est décrite par la structure "vif" suivante :

```
struct vif {
    u_char v_flags; /* VIFF_ flags defined above */
    u_char v_threshold; /* min ttl required to forward on vif*/
    u_int v_rate_limit; /* max rate */
    struct tbf *v_tbf; /* token bucket structure at intf. */
    struct in_addr v_lcl_addr; /* local interface address */
    struct in_addr v_rmt_addr; /* remote address (tunnels only) */
    struct ifnet *v_ifp; /* pointer to interface */
    u_long v_pkt_in; /* # pkts in on interface */
    u_long v_pkt_out; /* # pkts out on interface */
    u_long v_bytes_in; /* # bytes in on interface */
    u_long v_bytes_out; /* # bytes out on interface */
    struct route v_route; /* cached route if this is a tunnel */
    u_int v_rsvp_on; /* RSVP listening on this vif */
    struct socket *v_rsvpd; /* RSVP daemon socket */
};
```

La variable globale "viftable" est un tableau contenant la liste des interfaces virtuelles (vif). Un indice vers ce tableau est stocké dans la variable vifi_t, un entier cours non signé.

Dans notre cas, l'architecture SSM sur laquelle nous travaillons ne supporte pas les tunnels multicast. Un datagramme multicast ne peut être transmis qu'a une interface physique. Cependant une installation par le demon igmp-proxy de la liste des interfaces virtuelles au niveau du noyau est nécessaire. Nous avons effectué cette installation grace à l'appel système setsockopt() utilisé, cette fois ci, avec l'option MRT_ADD_VIF. Cette option communique directement avec le noyau en lui demandant d'exécuter la fonction add_vif disponible dans le fichier /usr/src/sys/netinet/ip_mroute.c . Cette fonction ajoute à la liste des interfaces virtuelles l'interface passée en argument de l'appel setsockopt. L'appel se fait comme suit :

```
if (setsockopt(socket, IPPROTO_IP, MRT_ADD_VIF,(char *)&vc, sizeof(vc)) <0)
    perror("setsockopt - MRT_ADD_VIF");
```

où vc correspond à l'argument de l'option MRT_ADD_VIF de l'appel setsockopt. Il s'agit d'une structure du type "struct vifctl". Cette strcture est definie dans le fichier /usr/include/netinet/ip_mroute.h comme suit :

```
struct vifctl {
  vifi_t  vifc_vifi;          /* index de l'interface virtuelle à ajouter */
  u_char  vifc_flags;         /*flag pour preciser le type de l'interface (physique/tunnel)*/
  u_char  vifc_threshold;     /* tt min nécessaire pour transmettre sur cette vif */
  u_int   vifc_rate_limit;          /* débit max */
  struct  in_addr vifc_lcl_addr;    /*adresse locale de l'interface */
  struct  in_addr vifc_rmt_addr;   /* addresse tunnel (non utilsée dans notre cas) */
};
```

L'argument "vc" doit être initialisé en affectant à chaque champ sa valeur corresondante.

6.3.3 Mise à jour de la MFC

La MFC est définie dans le fichier /usr/src/sys/netinet/ip_mroute.h comme suit :

```
struct mfc {
    struct in_addr mfc_origin;          /* IP origin of mcasts */
    struct in_addr mfc_mcastgrp;        /* multicast group associated*/
    vifi_t mfc_parent;                  /* incoming vif */
    u_char mfc_ttls[MAXVIFS];           /* forwarding ttls on vifs */
    u_long mfc_pkt_cnt;                 /* pkt count for src-grp */
    u_long mfc_byte_cnt;                /* byte count for src-grp */
    u_long mfc_wrong_if;                /* wrong if for src-grp */
    int  mfc_expire;                    /* time to clean entry up */
    struct timeval mfc_last_assert; v   /* last time I sent an assert*/
    struct rtdetq *mfc_stall;           /* q of packets awaiting mfc */
    struct mfc *mfc_next;               /* next mfc entry */
};
```

Plus simplement, on remarque que la structure mfc fait correspondre à chaque ensemble {adresse de source, adresse de groupe, adresse de l'interface entrante} la liste des interfaces sortantes pour lesquelles il faut transmettre les paquets multicast. Cette liste des interfaces sortantes est disponible dans le champ "mfc_ttls[MAXVIFS]". Ce champ contient pour chaque interface, identifiée par son index, le TTL (time to live) avec lequel les paquets multicast provenant de la source mfc_origin destinés au groupe mfc_mcastgrp et entrant de l'interface mfc_parent doivent être transmis. Pour que les paquets multicast soient transmis à une interface, le TTL correspondant à cette interface (mfc_ttls[vifi] avec vivi l'index de l'interface) doit être strictement positif. De plus, le TTL du paquet multicast doit être supérieur ou égal au TTL de l'interface sortante.

Il est clair donc que la décision de transmettre ou non un paquet sur une interface se traduit par la mise à zéro ou non du champ mfc_ttls[vifi]. Dans le cas ou la décision est de transmettre les datagrammes multicast, on choisit un TTL égal à 1. Avec ça, on est sûr

que tous les paquets, et quelque soit leur TTL, sont transmis.

Le mécanisme de redirection des paquets multicast est schématisé dans la figure 6.4. Les paquets IP sont reçus par le noyau au niveau de la file "Packets IP". Ceux-ci sont traités d'abord pas la routine ip_intr(). Cette routine délivre les paquets aux routines de traitement correspondantes selon leur type. Les paquets IGMP sont passés au démon IGMP-routeur (igmppd) tandis que les paquets multicast sont reçus et traités par la routine de redirection multicast ip_mforward(). Cette routine effectue une recherche dans la MFC (MFC lookup) de l'entrée (S,G) où S est la source et G est le groupe multicast. Cette entrée existe nécessairement puisque le paquet multicast ne peut être reçu par le proxy que si celui-ci a exprimé son désir de recevoir le trafic provenant de la source S destiné au groupe G. Une fois l'entrée (S,G) trouvée, la routine ip_mforward() consulte les valeurs du TTL correspondantes à cette entrée pour chaque interface. Ainsi, pour chaque interface, si le TTL est nul, le paquet n'est pas envoyé, s'il est égal à 1, le paquet est envoyé.

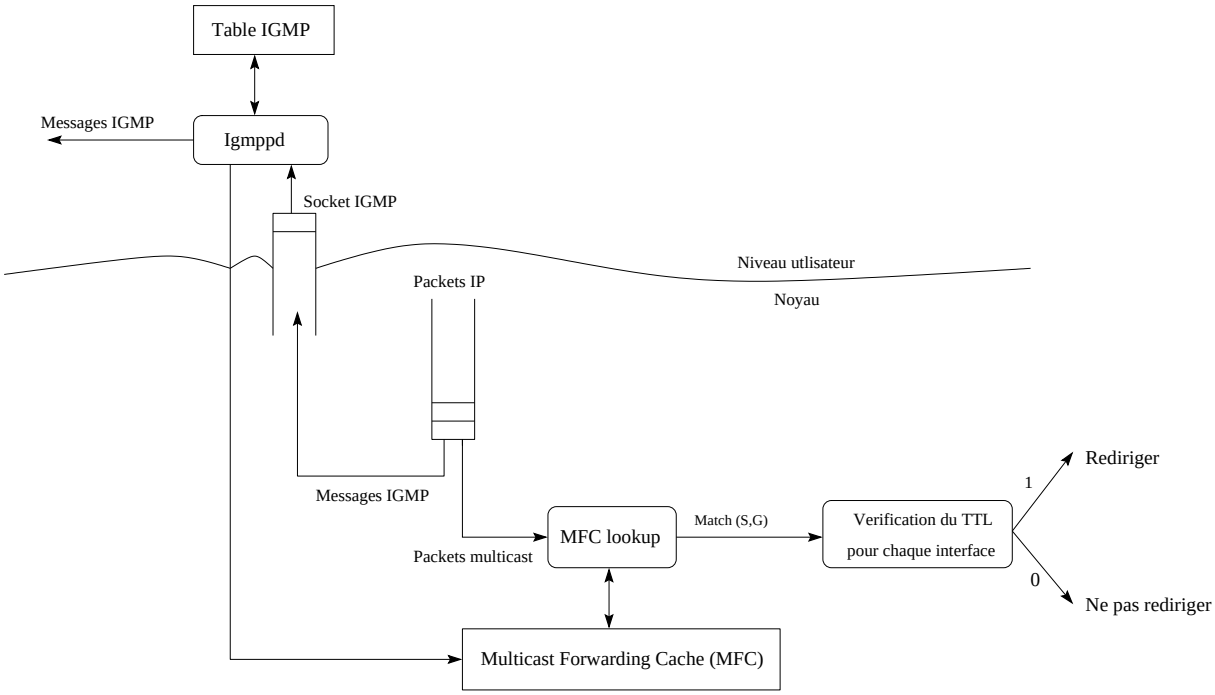

FIG. 6.4 – *Mécanisme de redirection des paquets Multicast par le Proxy IGMP*

Un timer est conservé pour chaque source. Il est mis à jour selon le type de report reçu et le mode de filtrage du groupe. Si le timer est strictement positif et le mode de filtrage est INCLUDE alors il y a un ou plusieurs systèmes qui désirent recevoir du trafic de cette source. Quand le timer expire, le routeur conclut qu'aucun système ne désire recevoir du trafic de cette source. Dans ce cas l'enregistrement de ce groupe sera effacé. Si le mode de filtrage est EXCLUDE, et le timer est strictement positif alors au moins un système désire recevoir du trafic de cette source. Si le timer expire dans ce mode alors le routeur informe le protocole de routage qu'aucun récepteur ne désire recevoir du trafic de cette source. La figure 6.5 représente les actions à prendre par le routeur selon les différentes valeurs de ce timer et selon le mode de filtrage du groupe. Elle résume les décision effectuées par le demon igmppd et transférées à la MFC sous forme de TTL.

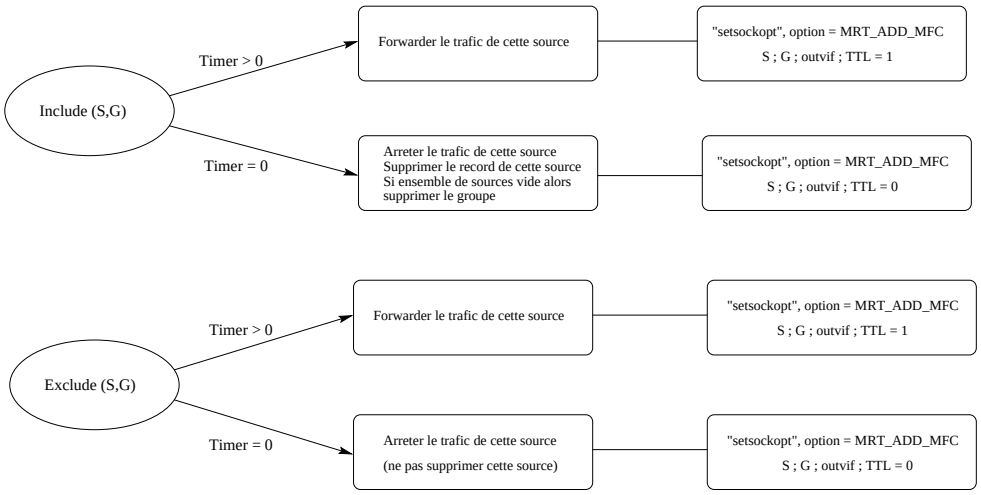

FIG. 6.5 – *Algorithme de gestion du timer d'une source*

Pour mettre à jour la MFC à chaque besoin, nous avons dû utiliser l'appel système setsockopt() avec l'option MRT_ADD_MFC. Cet appel accède directement à la fonction add_mfc() dans le noyau (fichier /usr/src/sys/netinet/ip_mroute.c) responsable de la modification d'une entrée de la MFC ou, le cas échéant, l'ajout d'une nouvelle entrée. La mise à jour de la MFC s'impose à chaque ajout d'une nouvelle source à la liste des sources d'un groupe (Timer >0) et aussi après l'expiration se son timer (Timer = 0).

L'appel setsockopt nécessaire se fait comme suit :

```
if (setsockopt(socket, IPPROTO_IP, MRT_ADD_MFC, (char *)&mc, sizeof(mc)) <0)
  perror("setsockopt - MRT_ADD_MFC");
```

où mc correspond à l'argument de l'option MRT_ADD_MFC de l'appel setsockopt. Il s'agit d'une structure du type "struct mfcctl". Cette strcture est definie dans le fichier /usr/include/netinet/ip_mroute.h comme suit :

```
struct mfcctl {
   struct in_addr  mfcc_origin;        /* adresse de la source */
   struct in_addr  mfcc_mcastgrp;      /* adresse multicast du groupe*/
   vifi_t  mfcc_parent;          /* interface entrante (inutile)*/
   u_char  mfcc_ttls[MAXVIFS];   /* valeur du ttl pour chaque interface */
   vifi_t  mfcc_oif;             /* index de l'interface sortante */
};
```

L'argument "mc" doit être initialisé en affectant à chaque champ sa valeur corresondante.

Remarquons que la structure mfcctl originale ne contient pas le champ "mfc_oif". Nous avons du l'ajouter pour préciser au noyau l'index de l'interface sortante concernée. En effet, à chaque appel de la fonction setsockopt() avec l'option MRT_ADD_MFC, nous avons besoin de changer dans la MFC la valeur du TTL de l'interface concernée seulement et ne pas changer sa valeur pour les autres interfaces. Pour cette raison, nous avons du introduire quelques modifications à la fonction add_mfc() dans le noyau.

Avant de changer le code, nous avons introduit une nouvelle option "IGMP_PROXY" dans le fichier de configuration du noyau disponible dans le répertoire "/usr/src/sys/i386/conf".

Les instructions initiaux permettant de changer la valeur du TTL sont les suivantes.

```
for (i = 0; i <numvifs; i++)
  rt->mfc_ttls[i] = mfccp->mfcc_ttls[i];
```

Nous avons changé ces instructions par ce qui suit :

```
#ifdef IGMP_PROXY
    rt->mfc_ttls[mfccp->mfcc_oif] = mfccp->mfcc_ttls[mfccp->mfcc_oif];
#else
    for (i = 0; i <numvifs; i++)
        rt->mfc_ttls[i] = mfccp->mfcc_ttls[i];
#endif
```

Ainsi, dans le cas ou l'option IGMP_PROXY est définie, seule le valeur du TTL de l'interface concernée sera changé. Les autres ne sont pas modifiés. Dans le cas contraire, le noyau executera les instructions initiaux en changeant touts les TTL.

Par ailleurs, il est important de noter que le mécanisme de redirection des paquets multicast odopté par le noyau nécessite la connaisssnce de l'interface entrante pour envoyer un paquet sur une interface. En effet, avant d'envoyer un datagramme, le noyau compare l'indice de l'interface delaquelle le paquet est arrivé avec celui disponible dans la MFC (cham "mfc_parent"). Seulememt en cas d'égalité, le paquet est transmis aux interfaces sortantes correspondantes. Cependant, le démon IGMP Proxy n'a pas de vision sur l'interface entrante de laquelle arrivent les paquets multicat. Pour cette raison, nous avons dû introduire des modifications au noyau et ce au niveau de la fonction "ip_mdq()" chargée de l'envoi des paquets multicast. Ces modifications ont pour but de permettre au noyau de rediriger les paquets en vérifiant seulememt les valeurs du TTL de chaque interface sortante et pas la valeur de l'interface entrante.

Dans l'impementation de la fonction "ip_mdq()", un test se fait pour vérifier la valeur de l'interface entrante. Si l'interface n'est pas la bonne, le paquet ne sera pas envoyé sur l'interface sortante. Nous avons du simplement elliminer ce test comme suit:

```
#ifdef IGMP_PROXY
    /* ne pas vérifier l'interface entrante */
```

```
    if (vifi >= numvifs) {
#else
    /* fausse interface entrante */
    if ((vifi >= numvifs) || (viftable[vifi].v_ifp != ifp)) {
#endif
```

6.3.4 Désactivation du processus de routage multicast

A l'arrêt du proxy, il est nécessaire de désactiver le processus de routage multicast afin de vider le liste des interface virtuelles et la MFC. Ceci permettra à un autre processus de routage multicast éventuel de fonctionner. La desactivantion du routage se fait par l'appel système "setsockopt" avec l'option 'MRT_DONE" comme suit :

```
int v = 0;
if (setsockopt(socket, IPPROTO_IP, MRT_DONE, (char *)NULL, 0) <0)
   perror("setsockopt - MRT_DONE");
```

6.4 Conclusion

Dans ce chapitre, nous avons décrit les principales étapes pour implémenter un proxy IGMP à partir de celle du protocole IGMP routeur. Ainsi, le proxy sera capable de rediriger les paquets multicast sans implémenter un protocole de routage multicast. Dans le chapitre suivant nous allons présenter une évaluation des performances de notre implémentation en comparant les mécanismes de routages adoptés pas le proxy IGMP avec celle qu'introduit le protocole PIM-SM

D'autre part, cette implémentation n'introduit pas de mécanismes de sécurité. Il reste donc à integrer le protocole BAAL au proxy, en l'occurance la partie controleur local de ce protocole.

Chapitre 7

Evaluation des performances du proxy IGMP

7.1 Introduction

Après avoir détaillé l'implémentation du proxy IGMP, il s'avére nécessaire d'évaluer ses parformances. Dans ce chapitre, nous présenterons les tests que nous avons réalisé pour vérifier le fonctionnement du proxy ainsi qu'une évaluation qualitative des performances du proxy en le comparant avec un routeur multicast classique implémentant PIM-SM et IGMPv3.

7.2 Test du Proxy IGMP

Pour tester le Proxy IGMP, nous avons utilisé une plate-forme de test. Cette plate-forme est constituée de PC sur lesquels nous avons installé FreeBSD 4.5. (figure 7.1). Nous avons installé sur toutes ces machines la partie hôte de IGMPv3. Le Proxy et la partie routeur de IGMPv3 sont déployés sur les machines sous forme d'un démon "igmppd". Pour réaliser les tests nous avons utilisé l'outil TcpDump[1] . L'utilisation de cet outil nous

1. TcpDump est un outil de visualisation des paquets IP. Sa nouvelle version permet de visualiser les messages IGMPv3.

permet de visualiser les messages IGMPv3 ainsi que les paquets multicast. Après la mise en place de cette plate-forme, nous avons essayé des petites applications multicast en choisissant un hôte comme source de données et un autre comme récepteur. Ceci pour voir les performances du Proxy IGMP que nous avons implémenté.

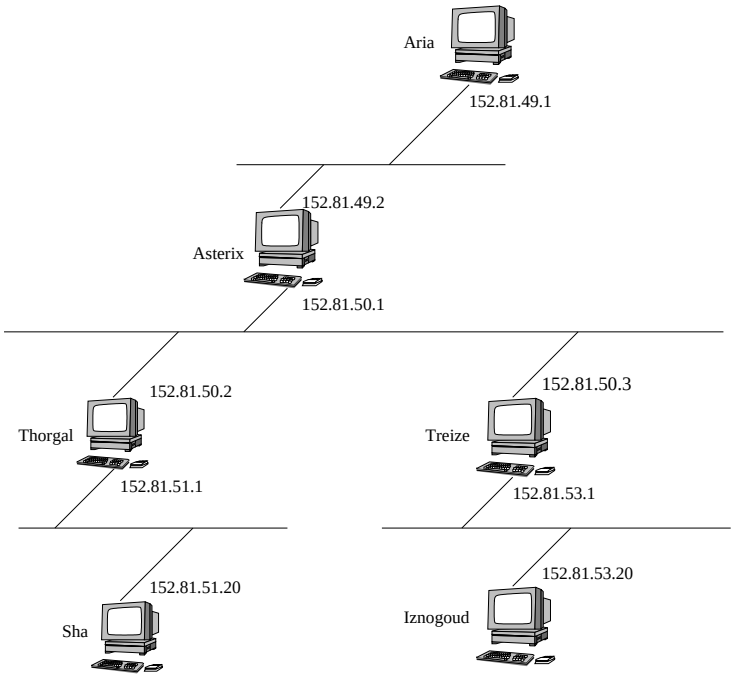

FIG. 7.1 – *Plateforme de test*

Le premier test que nous avons réalisé était de lancer "IGMP proxy" sur Asterix et Treize en configurant 152.81.49.2 comme interfce upstream de Asterix et 152.81.50.3 comme interface upstream de Treize. Nous avons aussi lancé "mrcvie"[2] sur Iznogoud et "msend"[3] sur Aria. Lorsque Iznogoud, grâce à l'application "msend" joint un groupe donnée avec un mode de filtrage Include et une source 152.81.49.1 (Aria), on remarque bien qu'il reçoit les paquets envoyés par Aria desinés à ce groupe. Par contre lorsque le mode est Exclude (152.81.49.1), les proxy intermédiaires ne transmettent pas les paquets envoyés

2. "mrcvie" est une application multicast permettant à un hôte de joindre un groupe et spécifier le mode de filtrage et la liste de sources"

3. "msend" est une application qui permet d'envoyer des paquets à un groupe multicast sur un port spécifié

par Aria. Ainsi le proxy IGMP joue bien son rôle lorsque la source est liée à l'inteface upstream. D'autre part, nous avons testé le cas ou la source est sur une interface downstream et les membres sont répartis sur les autres interfaces downstream et l'interface upstream. Nous avons bien vérifié que le proxy IGMP joue parfaitement son rôle en redirigeant les paquets multicast aux membres demandant le trafic multicast d'un groupe donné, qu'ils soient adhérés sur l'interface upstream ou sur l'une des interfaces downstream.

D'autre part, Il est important de noter que déploiement du proxy IGMP exige qu'il puisse communiquer avec les routeurs PIM-SM. En effet, une architecture multicast courante mettant en œuvre le proxy IGMP se compse d'un réseau LAN désservi par un routeur d'accès PIM-SM directement connecté à Internet suivi d'un proxy IGMP. La source se trouve quelque part dans le réseau Internet. Pour cette raison, nous avons dû tester les proxy IGMP lorsqu'il est directement lié à une routeur implémentant PIM-SM et IGMPv3. Nous avons lancé PIM-SM et IGMPv3 sur Asterix, IGMP proxy sur Treize, "mrcvie" sur Iznogoud et "msend" sur Aria. Nous avons bien vérifié que les paquets multicast arrivent au membre Aria lorsqu'il les demande (include 152.81.49.1) et n'arrivent pas dans le cas contraire.

7.3 Comparaison des performances du proxy IGMP avec celles d'un routeur PIM

7.3.1 nombre de messages

Dans le premier chapitre, nous avons bien détaillé le fonctionnement du protocole de routage mlulticast PIM-SM. En particulier, nous avons présenté les différents messages qu'introduit PIM pour assurer la redirection des paquets multicast. Ces messages sont au nombre de 5 et sont les messages join, prune, register, register stop et assert. Ces messages s'agoutent aux messages IGMP usuels pour permettre à un routeur multicast classique de fonctionner. Cependant le proxy IGMP que nous avons implémenté ne met en œuvre aucun de ces messages PIM. Il n'utilise que les messages IGMP et exploite au maximum l'information receuillie par ce protocole (IGMP) pour rediriger les paquets

multicast. Ainsi, il est clair qu'un nombre de messages nettement inférieur permet un encombrement minumum du réseau ainsi qu'un temps de traitement bien plus faible.

7.3.2 mécanisme de redirection au niveau noyau

Pour pouvoir comparer les performances du mécanisme de redirection des paquets multicast entre un routeur PIM et un proxy IGMP, il conviant de détailler celui adopté par PIM-SM [HEL96](figure 7.2)

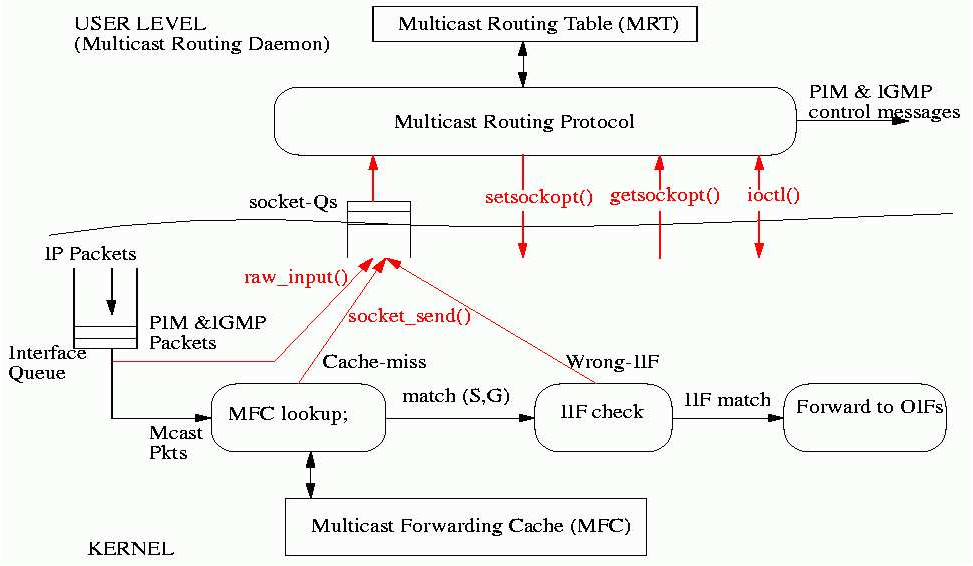

FIG. 7.2 – *Mécanisme PIM de redirection des paquets multicast*

Le démon PIM traite tous les messages de contrôle PIM et a installe un environnement approprié au niveau du noyau pour livrer des paquets multicast. Le noyau doit supporter l'expédition de paquets multicast.

Quand le noyau reçoit un paquet IP, il le passe par la routine de réception des paquets IP [ip_intr()]. ip_intr délivre le paquet aux routines de traitement appropriées en se référant à l'adresse de destination et le numéro du protocole. Trois cas peuvent se présenter : Le datagramme IP est un paquet IGMP, PIM ou un paquet multicat.

Dans le cas où il s'agit d'un paquet multicast, celui-ci est traité par la routine d'expédition multicast au noyau [ip_mforward()]. Par la suite, cette routine consulte la

MFC pour y chercher l'entrée (S,G) correspondante aux adresses de groupe et de source multicast en question. Dans le cas où l'entrée existe (match(S,G)), une verification de l'interface entrante (iif) à lieu. S'il y a concordance (IIF match), le paquet est expédié aux interfaces sortantes correspondantes. (oifs) Cependant, dans le cas où l'entrée (S,G) n'a pas été trouvée dans la MFC (Cache-miss) ou l'interface entrante n'est pas la bonne (Wrong IIF), un message de contrôle interne est envoyé au démon pour être traité.

En revanche, le proxy IGMP ne fait pas intervenir tous ces mécanismes. En particulier le phénomène de "Cache-miss" ne peut jamais avoir lieux. En effet, le proxy IGMP reçoit un paquet multicast seulement lorsqu'il a demandé de le recevoir en s'adhérant sur son interface upstream aux groupes auxquels l'une de ses interface downstream est adhérée. Ainsi, l'entrée (S,G) est crée dans la MFC avant même que le proxy ne commence à recevoir les paquets multicast pour ce canal. Par ailleurs, rappelons que le phénomène "Wrong IIF" ne paut aussi avoir lieu dans un proxy IGMP et ce grâce aux modifications que nous avons dû apporter au noyau et qui sont expliquées dans le paragraphe 4.2.2.

7.3.3 Comparaison des démons PIM et IGMP-proxy

Le démon PIM [HEL96] effectue le traitement de tous les messages de contrôle internes (Cache-miss, Wrong IIF...). il effectue aussi le réception, l'envoi et le traitement des différents messages PIM (assert, join, prune, register). D'autre part, de démon construit la table de routage multicast qu'il utlisera pour contrôler la redirection des paquets multicast. Toutes des fonctionnalités et leur intéractions sont représentées dans la figure 7.3qui présente le diagramme de flow des données à l'interieur d'un routeur multicast classique (PIM-SM + IGMPv3).

Bien entendu, le démon du proxy IGMP ne fait intervenir ni table de routage multicast ni traitement des messages de contrôle internes et les messages PIM. Ainsi, le diagramme de flow de données du proxy IGMP ne contiendra ni la partie "accept_pim()" ni "process_upcall()" représentés dans la figure 7.3.

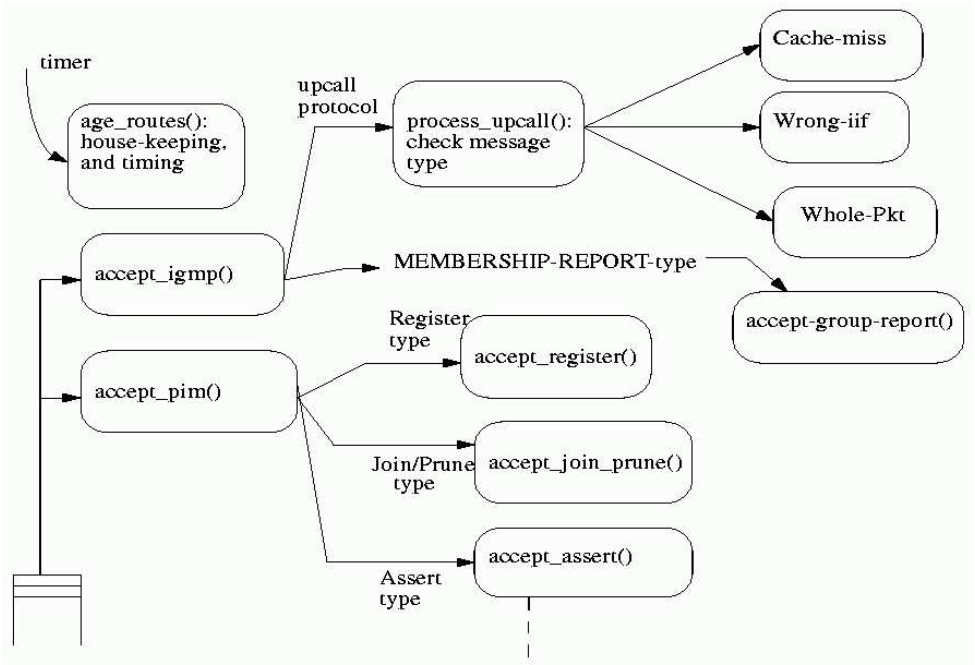

FIG. 7.3 – *Diagramme de flow de données dans un routeur PIM*

7.4 Conclusion

Suite à cette comparaison, nous parvenons à mettre en évidence l'intéret du proxy IGMP. En effet, il met en œvre un processus nettement plus optimisé que celui de PIM tant au niveau noyau qu'au niveau utlisateur. Il permet aussi une surcharge minimale du réseau en évitant tous les messages PIM. On peut ainsi déduire que le temps de traitement de de redirection que met un proxy IGMP pour transmettre un paquet multicast est nettement inferieur à celui mis par un routeur multicast classique.

Chapitre 8

Conclusion générale

L'objectif de ce stage était d'implémenter un proxy IGMP qui assure la redirection des paquets multicast sans protocole de routage.

La première partie du travail était de spécifier les principaux composants du proxy ainsi que ses différentes fonctionnalités. A l'issue de cette partie, nous avons aboutit a montrer que le proxy contient une interface upstream unique et plusieurs interfaces downstream. Il assure la partie routeur du protocole IGMP sur ces interfaces downstream et la partie hôte de ce protocole sur son interface upstreaum. D'autre part, nous avons dégagé que le proxy doit effectuer deux fonctions essentielles. La première fonction est la réception des paquets multicast demandés par les membres adhérés sur ses différentes interfaces. Pour ce faire, le proxy doit se comporter comme un hôte qui, en recevant une demande d'adésion sur l'une des ses interfaces downstream, envoie à son tour une demande d'adhésion pour le même groupe sur l'interface upstream et ainsi recevoir le trafic pour ce groupe. D'autre part, le proxy effectue le filtrage sur les sources en blockant le trafic des sources non désirées par les membres et permettre celui des sources désirées. D'autre part, le seconde fonctionnalité aussi importante que la première est la redirection des paquets multicast. Le proxy donne ses consignes au noyau pour controler le forwarding des datagrammes multicast en fonction de l'information disponible dans le table IGMP. Le proxy effectue ainsi les fonctions d'initialisation du processus de routage multicast, d'installation de la liste des interfaces virtuelles et enfin de maintien et de mise à jour de la MFC.

Après avoir implémenté et testé le proxy IGMP nous avons évalué ses performances. Ainsi, nous avons mis en évidence que le processus de routage exercé par le proxy est nettement allégé et optimisé par rapport à celui introduit par PIM-SM. Ces avantages apparaissent aussi bien au niveau du démon qu'au niveau du noyau.

La dernière partie de ce travail était d'intégrer le proxy au protocole BAAL. Ce protocole a été spécifié et implémenté au sein du projet RESEDAS. Baal est un protocole de distribution et de maintenance de clés de groupes; ce protocole entre dans le cadre de la sécurisation de la communication de groupes. Il fait intervenir trois entités distinctes qui sont le controleur de groupe, le controleur local et l'hote Baal. Mon intervention était au niveau du controleur local en lui intérant le proxy IGMP. Ainsi nous avons abouti à une implémentation complète des différents composants d'une architecture SSM sécurisée utilisant un proxy IGMP.

Ce stage m'a permis de découvrir le monde de la recherche, de m'intégrer à l'équipe RESEDAS et de me familiariser avec les travaux qu'elle mène actuellement. D'un autre côté, j'ai pu toucher de près à la programmation noyau avec le langage C sous le système d'exploitation FreeBSD, ce qui m'a permis de solidifier mes connaissances sur ce langage.

Je tiens aussi à dire, que c'était pour moi une grande occasion pour travailler sur un vrai système informatique en réseaux, où tout est accessible de partout avec transparence et fiabilité garanties.

éèêàùç

Bibliographie

[GRA00] W.de graaf, "A host Implementation of IGMPv3 on FreeBSD", http://home.hetnet.nl/~wilbertdg/igmpv3.html, Janvier 2000.

[LAH01] A. Lahmadi "mise en œuvre d'un outil de gestion de clés de groupes dynamiques" , rapport de projet de fin d'études, Pages 25-39 , Juin 2001

[HEL96] A.helmy, "Protocol Independent Multicast-Sparse Mode (PIM-SM): implementation document", http://netweb.usc.edu/pim/pimd/docs/implem-doc.ps , Aout 1996.

[CAI02] Brad Cain, Steve Deering, Bill Fenner, Isidor Kouvelas, Ajit Thyagarajan, "Internet Group Management Protocol, version 3" , IETF draft, IDMR working group , draft-ietf-idmr-igmp-v3-09.txt, Janvier 2002.

[FEN97] W.Fenner,"Internet Group Management Protocol, Version 2", RFC 2236, Novembre 1997.

[CHA99] Ghassen Chaddoud, Isabelle Chrisment, André Schaff, "Baal: sécurisation des communications de groupes dynamiques", 99.

[CRI01] Isabelle Crisment, Abdelkader Lahmadi, Gassan Chaddoud, "Implémentation d'un prototype du protocole Baal", Rapport de recherche, Decembre 2001.

[EST97] D.Estrin, D.Farinacci et A.helmy, "Protocol independant multicast sparse mode (pim-sm)", RFC-2117, juin 1997

[HAN95] M. Handley, V. Jacobson, "SDP: Session Description Protocol" , Internet draft, Novembre 1995,

[PUS98] T.Pusaterie, "Distance vector multicast routing protocol", Internet draft , Mars 1998

[FEN01] W.Fenner, "IGMP-based Multicast Forwarding (“IGMP Proxying”)", IETF draft, IDMR group : <draft-ietf-idmr-igmp-proxy-01.txt>, Juillet 2001

[DEE96] Stephen Deering, Ahmed Helmy, Mark Handley, “Protocol independent multicast-Sparse Mode : morivation and architecture”, IETF draft, <draft-ietf-idmr-pim-arch-04.ps>, October 1996.

[DIO02] Christophe Diot, John Meylor, David Meyer , Brian Heberman, “An overview of source-specefic multicast (SSM) deploiment” Internet draft : <draft-ietf-ssm-overview-02.txt>, Juin 2002.

[CHA02] Ghassan Chaddoud, Isabelle Crisment, André Schaff, “S-SSM : a setcure SSM archirecture” , IETF draft, <draft-chaddoud-sssm-00.txt>, Février 2002.